AD 745139

# Strain Distribution Around Underground Openings

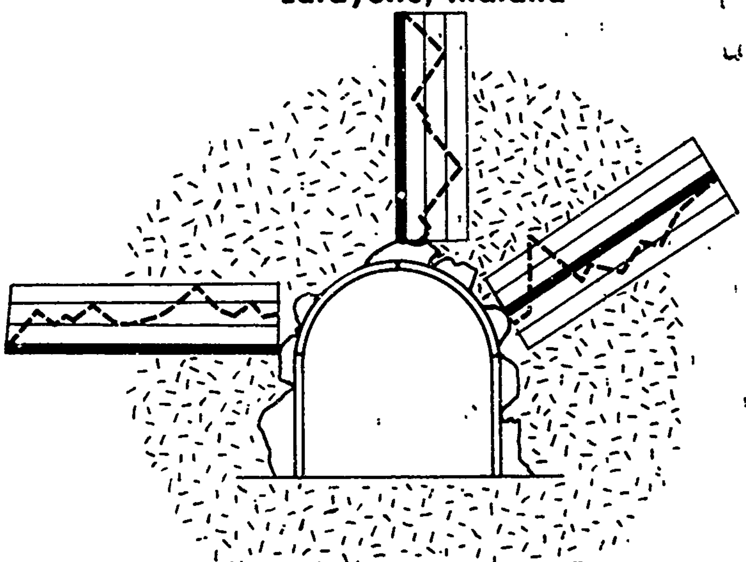
## Technical Report No. 7

## FINITE ELEMENT ANALYSIS OF JOINTED SYSTEMS


**W. H. Perloff**

Soil and Rock Mechanics Area
School of Civil Engineering
Purdue University
Lafayette, Indiana




Sponsor

ADVANCED RESEARCH PROJECTS AGENCY
DEPARTMENT OF DEFENSE


TR-6
AD 735376




MAY, 1972

Prepared For
OFFICE OF THE CHIEF OF ENGINEERS
DEPARTMENT OF THE ARMY
WASHINGTON, D.C.




152

Strain Distribution Around Underground Openings

Techn.  al Report No. 7

# FINITE ELEMENT ANALYSIS OF JOINTED SYSTEMS

W. H. Perloff

Soil  Rock Mechanics Area
School of Civil Engineering
Purdue University
Lafayette, Indiana

Contract No. DACA 73-68-C-0002(P002)

Sponsor

ADVANCED RESEARCH PROJECTS AGENCY
DEPARTMENT OF DEFENSE

May , 1972

Prepared For

OFFICE OF THE CHIEF OF ENGINEERS
DEPARTMENT OF THE ARMY
WASHINGTON, D.C.

## SUMMARY

This report describes progress on the analytical portion of project number DACA-73-68-C-002 for the period 1 April 1969 to 31 December , 1971.

The static SLAM finite element code was extended to include jointed systems with elastic-plastic mechanical characteristics satisfying a variety of possible yield criteria. The results indicated that

1. The two-dimensional static SLAM code for plane jointed systems could predict the response of such systems to imposed loadings when the system and loadings were properly characterized.
2. Predictions of displacements around excavations in natural jointed rock masses deviated from measured values. Reasons for this included both difficulty in correctly determining displacements in rock masses in the field as well as discrepancies between the simplified representation of natural conditions and the conditions themselves.
3. It was not practicable at the present time to carry out a three-dimensional finite element representation of excavations in natural jointed rock.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT'D)

# LIST OF FIGURES

## SECTION 1 - INTRODUCTION

### 1.1 - Introduction

This report describes the analytical portion of project no. DACA-73-68-C-0002 for the period 1 April, 1969 to 31 December, 1971. As a part of the total effort to determine if the element method analysis will permit prediction of safe spans for existing and proposed underground openings, it was the objective of this portion of the project to extend the continuum model for underground openings, employing the finite element method to consideration of the rock mass as a two-dimensional jointed medium, in order to improve prediction of stresses around and displacement within underground openings. Three-dimensional effects were considered also.

### 1.2 - Outline of Progress

Accomplishment of this objective is described in the following sections. These sections consider the following areas of progress:

1. Modification of the existing continuum static SLAM Finite Element Code to incorporate direct consideration of joints and joint systems normal to the plane problem.
2. Comparison of analysis with experiments conducted on a small scale jointed model of a simulated rock system.
3. Results of analysis of an underground opening considering the influence of the observed joint systems, and preliminary comparisons with measured displacements.
4. Consideration of extension to three dimensions.

## SECTION 2 - MODIFICATION OF STATIC SLAM CODE TO INCORPORATE JOINTS

### 2.1 - Review of Static SLAM Code

The static SLAM (Stresses In Layered Arbitrary Media) Code was developed for finite element analysis of large systems of continuous media. An outgrowth of the code for dynamic problems developed at IITRI by Costantino (1966, 1968) and Wachowski and Costantino (1966), its salient features are described by Perloff (1969).

The Static SLAM Code is characterized by a number of features which distinguish it from other available codes. These include:

1. The code contains an algorithm for renumbering the nodes so that the minimum band width of non-zero terms within the stiffness matrix results. This leads to an efficient operation, especially for large problems. Furthermore the user is able to number node points arbitrarily. Details are discussed by Wachowski and Costantino (1966).
2. A non-linear displacement field is assumed for rectangular elements, so that where rectangular elements can be incorporated in the geometry, fewer elements are required to represent the problem (Costantino, 1966).
3. The constitutive laws used in the code are contained within a material "catalog" and new constitutive relations can be added without modifying the basic code.

Revised solution procedure

Two revisions have been incorporated in the solution procedure for the code to reduce computer time:

1. The initial elastic solution is obtained by direct elimination of the node point equilibrium equations, rather than by iteration as in the earlier version of the code (Perloff, 1969).
2. An over-relaxation factor is incorporated in the iterative procedure for determining the node point displacements when all or part of the continuum is behaving in a non-linear fashion. When it is determined that the yield point has been exceeded in one or more elements, the applied boundary loadings and displacements are reduced until all elements are acting in the elastic range. The remaining nonlinear part of the solution is carried out in a series of small steps, by increasing the applied loads or displacements in increment until the final loading condition is reached. At each increment the node point displacements and loads are determined and added to those from the previous increment. For each nonlinear increment the initial trial solution for the iteration is the displacement field obtained from the previous increment. For the first nonlinear increment, the elastic solution is used as the initial trial solution.

The system equilibrium equations for the nonlinear increment at each node are:

$$[K]\{\Delta U\} = \{\Delta R\} + \{\Delta R^N\} \tag{2.1}$$

in which [K] is the stiffness matrix of the continuum composed of the assembled elements, calculated by adding the stiffnesses of all elements in the system, $\{\Delta U\}$ are the node point displacements, $\{\Delta R\}$ are the applied node point loads and $\{\Delta R^N\}$ are the incremental nonlinear correction terms in the applied node point loads. The error at each

stage of the iteration process is then

$$\{\Delta\}^i = [K]\{\Delta U\}^i - \{\Delta R\} - \{\Delta R^N\} \tag{2.2}$$

in which the superscript i indicates the $i^{th}$ iteration. The displacement increment for the (i+1) increment is then

$$\{\Delta U\}^{i+1} = \{\Delta U\}^i - \alpha\{\frac{\Delta}{K_m}\}^i \tag{2.3}$$

where $K_m$ is the main diagonal stiffness at node m, and $\alpha$ is the over-relaxation factor. The iteration process is carried out until the specified allowable error is reached at each node.

## 2.2 - Behavior of Joints

Most natural rock contains more or less planar surfaces across which the rock has separated at some time in the past. Such defects, called joints, commonly occur as approximately parallel multiple surfaces spaced from fractions of an inch to many feet apart. Systems of joints frequently intersect so that a large rock mass may contain many such families at various spacings and orientations. It is generally recognized that the mechanical behavior of masses of rock is influenced strongly by the presence of such joint systems, along with other geologic defects; and this has been demonstrated by field observations and laboratory experiments (Obert, 1967; Rosenblad, 1971).

Joints may be clean surfaces of separation, or they may be filled with a variety of materials. Sometimes joints contain precipitates, such as calcite or chlorite, which may have a strength approximately the same as that of the natural rock and which may serve as cementing agents to impart tensile resistance normal to the joint. Other filling materials such as clays, lead to joints which are much weaker than the intact rock. In the

case of unfilled joints, the rock on either side of the joint is frequently altered to a weaker, or less stiff form by chemical and/or mechanical action.

Natural joint surfaces are rarely smooth. Even when they are approximately planar, they contain asperities which impart roughness to the joint. The role of these asperities in the shearing resistance along joints is a function of the magnitude of the pressure normal to the joints (Patton, 1966).

Because of the approximately planar nature of most joint systems it is useful to describe the mechanical behavior of joints in terms of stresses and displacements normal to and parallel to the joint surface.

The relationship between the average shear stress applied to a joint and the shear displacement, or strain, corresponding to a given normal pressure can be idealized as shown in Figure 2.1. That is, the joint deforms in a more or less linear way until the yield, or peak strength is reached. Further displacement occurs at a shear stress magnitude equal to that of the residual strength. The residual strength is usually equal to or less than the peak value.

The magnitude of the peak strength has been commonly described in terms of the normal stress on the joint by the conventional two-dimensional Mohr-Coulomb criterion

$$\tau_f = c + \sigma_f \tan \phi \qquad (2.4)$$

in which $\tau_f$ is the shear stress on the joint at failure, c is the magnitude of the peak stress at zero normal pressure on the joint, $\sigma_f$ is the normal pressure acting on the joint, and $\tan \phi$ is the slope of the shear strength envelope illustrated in Figure 2.2a. Recent evidence, (Patton, 1966; Rosenblad, 1971) suggests that a bilinear relation for the peak stress,

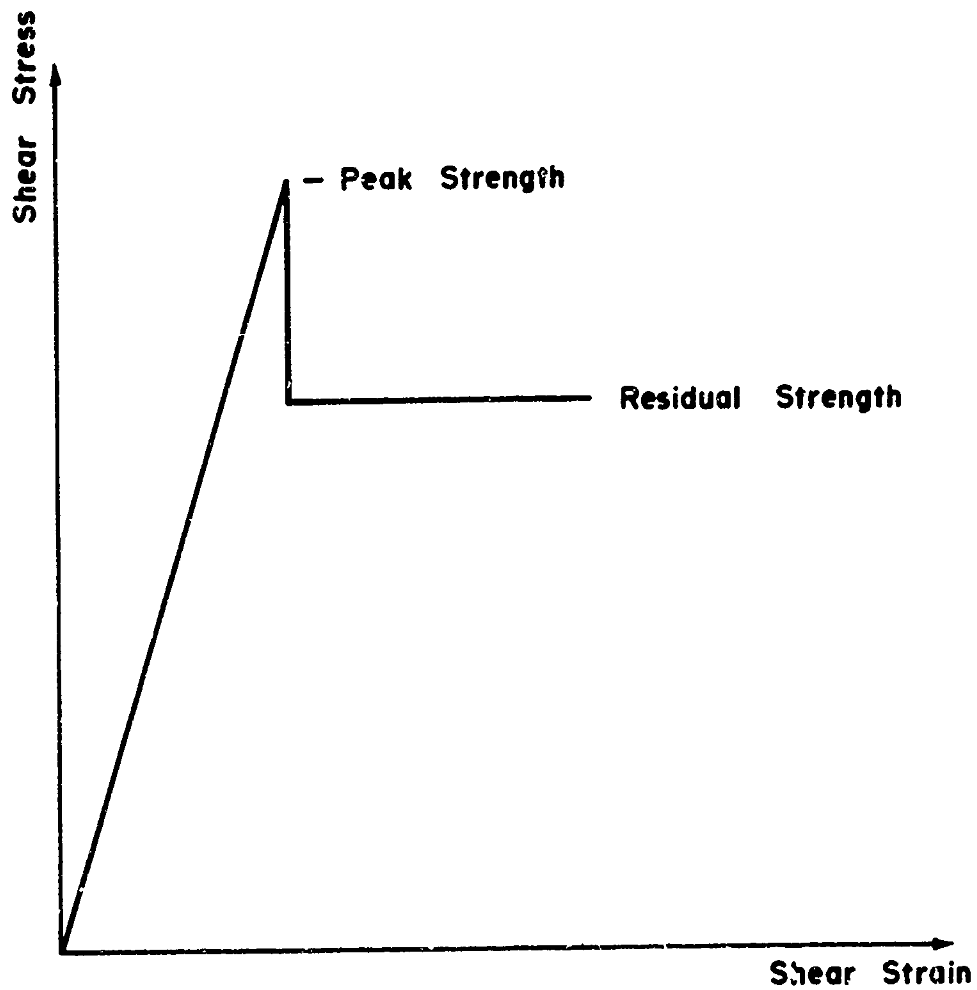
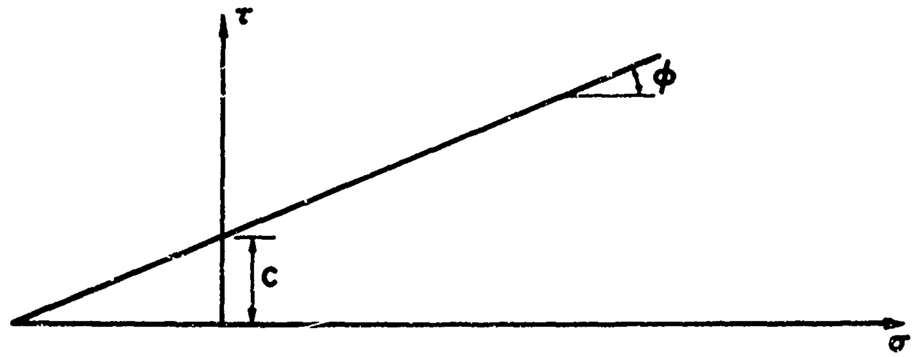
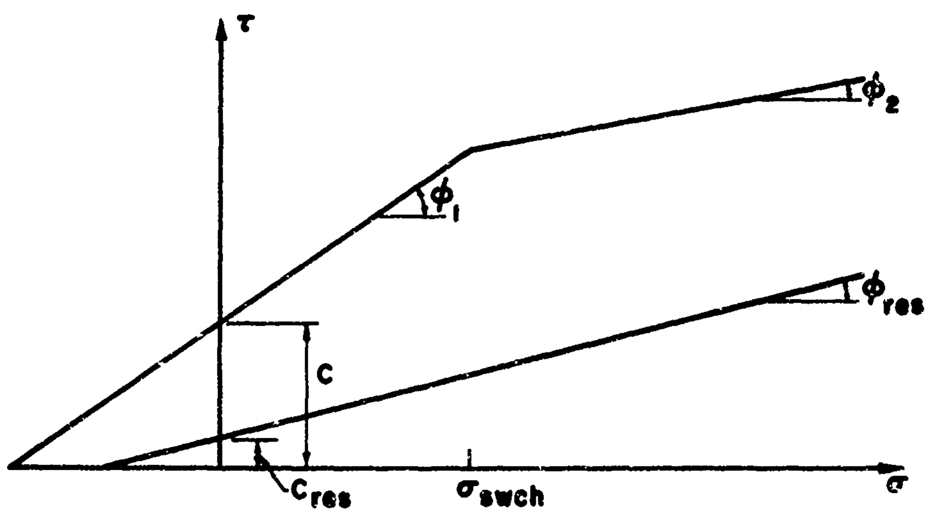



Figure 2.1 - Shear Stress - Shear Strain Relation for Two - Dimensional Mohr - Coulomb Material




a) - Linear Mohr - Coulomb Criterion for Peak Strength




b) - Bilinear Mohr - Coulomb Criterion for Peak Strength with Lower Residual Strength

Figure 2.2 - Yield Criteria for Joint Elements

as illustrated in Figure 2.2b is more appropriate.

$$\tau_f = c + \sigma_f \tan \phi_1 , \qquad \sigma_f \leq \sigma_{swch} \tag{2.5a}$$

$$\tau_f = c + \sigma_{swch} \tan \phi_1 + (\sigma_f - \sigma_{swch}) \tan \phi_2 , \quad \sigma_f > \sigma_{swch} \tag{2.5b}$$

in which $\sigma_{swch}$ is the normal stress at which the bilinear failure envelope changes slope.

The residual shear strength relation, also shown in Figure 2.2b is

$$\tau_{f_{res}} = c_{res} + \sigma_f \tan \phi_{res} \tag{2.6}$$

in which the subscript (res) denotes the residual shear strength parameters.

While Figure 2.1 and Equations 2.5 and 2.6 constitute a somewhat idealized depiction of the observed behavior of natural and artificially created joints, the difficulty of testing insitu joint behavior and the variability of results (Goodman, 1969) suggest that the above description is sufficiently detailed at the present time.

## 2.3 - Joint Elements

In the Static SLAM Code joints are characterized as rectangular elements of zero thickness. This is illustrated in Figure 2.3 which shows a rectangular element of length a and width b in the plane of the page. The joint is described by such an element in which the dimension b approaches zero so that node points i and $\ell$ have the same coordinates, and node points j and k have the same coordinates. An elongate joint is then made up of a series of such joint elements to which suitable elastic-plastic properties have been assigned.

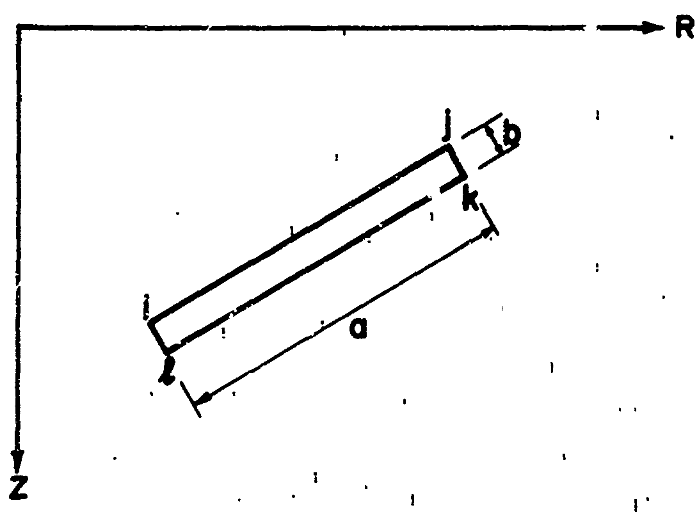



Figure 2.3 - Rectangular Element

The stiffness matrix for the joint elements is derived by determining the stiffness for a rectangular joint in terms of the width b and then allowing b to approach zero in the limit. That is, an equivalent strain, $\{\varepsilon'\}$ is defined as

$$\{\varepsilon'\} = b\{\varepsilon\} \tag{2.7}$$

in which $\{\varepsilon\}$ is the appropriate strain vector. The stresses, $\{\sigma\}$ are

$$\{\sigma\} = [C'] \{\varepsilon'\} \tag{2.8}$$

where

$$[C'] = \frac{1}{b} [C] \tag{2.9}$$

and [C] is the matrix of elastic constants.

Imposing a set of virtual nodal displacements $\{\delta u\}$, the equivalent strain is related to the virtual node point displacements by

$$\{\delta\varepsilon'\} = b[A] \{\delta u\} = [A'] \{\delta u\} \tag{2.10}$$

where the matrix [A] is determined from the definition of the strain components and the assumed displacement field for the element. The internal strain energy $\delta W_i$ developed by these displacements is

$$\delta W_i = \frac{1}{b} \int_V \{\delta\varepsilon'\}^T\{\sigma\} \, dV \tag{2.11}$$

in which the superscript T denotes the transpose of the matrix and the integration is taken over the volume V of the element. The corresponding external work done by the node point resisting forces during the virtual displacement $\delta W_e$ is

$$\delta W_e = \{\delta u\}^T\{S\} \tag{2.12}$$

in which {S} is the vector of node point forces for the element.

Invoking the principal of virtual work, expressions 2.11 and 2.12 are equated. Substituting Equation 2.10 into the result yields

$$\{\delta u\}^T\{S\} = \frac{1}{b} \int_V \{\delta\varepsilon'\}^T \sigma \, dV$$

$$= \frac{1}{b} \int_V \{\delta u\}^T[A']^T \sigma \, dV \tag{2.13}$$

Or, solving for the node point forces,

$$\{S\} = \frac{1}{b} \int_V [A']^T\sigma \, dV$$

$$= \frac{1}{b} \int_V [A']^T[C'][A'] \, dV \; \{u\} \tag{2.14}$$

This can be written

$$\{S\} = [k] \; \{u\} \tag{2.15}$$

in which the stiffness matrix [k] is

$$[k] = \int_V \frac{1}{b} [A']^T[C] \frac{1}{b} [A'] \, dV \tag{2.16}$$

When the rectangular element is a joint element, the stiffness is then

$$[k]_{joint} = \lim_{b \to 0} [k] \tag{2.17}$$

The individual terms of a stiffness matrix which are preserved are determined by substituting the appropriate element integrals as given by Costantino and Wachowski (1966).

## 2.4 - Constitutive Laws for Joint Elements

Three constitutive laws are provided in the SLAM Code material catalog for the description of the mechanical behavior of the joint elements.

They are:

1. An elastic-plastic material obeying the Von Mises yield criterion and the Prandtl-Reuss flow equations. This model, which incorporates strain hardening effects, is described in detail by Perloff (1969) and Costantino (1968). The constitutive relation can be employed for regular elements as well as joint elements.
2. An elastic-plastic material obeying the Drucker and Prager (1952) three-dimensional extension of the Mohr-Coulomb criterion. This relation which can also be viewed as an extended Von Mises yield criterion is also described in the earlier report (Perloff, 1969) and by Costantino (1968). Although usable for both joint elements and regular elements, this constitutive relation is probably applicable to joints only when they are filled.
3. An elastic-plastic material which obeys a two-dimensional bilinear Mohr-Coulomb yield criterion described in Equations 2.5, and depicted graphically in Figure 2.2b. Post-yield behavior is governed by the residual strength parameters as indicated in Equation 2.6 and Figure 2.2b. An option is also provided in the SLAM code to require that the joint is incapable of withstanding tension normal to the joint surface. In Figure 2.2b this would correspond to a case in which the failure envelopes would be vertical along the $\tau$ axis. Such a case corresponds to a clean unhealed joint.

Plastic strains, for the post-yield condition, are calculated as,

$$\{\varepsilon^P\} = \{\varepsilon^T\} - \{\varepsilon^E\} \tag{2.18}$$

in which $\{\varepsilon^T\}$ are the total computed strains determined from the

node point displacements, and $\{\varepsilon^E\}$ are the elastic strains determined from

$$\{\varepsilon^E\} = [C]^{-1} \{\sigma\} \tag{2.19}$$

This constitutive relation is applicable to joint elements only.

These constitutive laws permit consideration of a wide range of types of joint behavior. The nature of the material catalog in the SLAM code also allows for relatively straightforward incorporation of additional constitutive relations, such as those involving time-dependent behavior.

## 2.5 - Revised SLAM Code

The current version of the Static SLAM Code, containing the revisions described above and incorporating consideration of joint elements is listed in Appendix A. The form of the data input required is given at the beginning of Appendix A and is indicated by comment cards within the code itself.

## 2.6 - Interpolation Code for MPBX Displacements

To assist in comparing the results of the analysis with displacement measurements along MPBX lines, an auxiliary code has been developed to compute displacements along these lines. The node point displacements determined by the SLAM code are used as input to the interpolation code for MPBX displacements. The input node point coordinates and displacements may be in either magnetic tape or punched card form.

Details of data input and a listing of the code are given in Appendix B.

# SECTION 3. RESULTS OF ANALYSES OF JOINTED SYSTEMS

## 3.1 - Introduction

Two types of jointed systems were selected for comparative analyses to indicate the degree to which the behavior of the system could be predicted by the static SLAM code incorporating joint elements. The first of these was a series of model tests conducted on a mass of simulated rock blocks arranged to provide two families of intersecting joints. This model was developed at the Missouri River Division Laboratory (MRDL) of the U. S. Army Corps of Engineers (Rosenblad, 1971).

The second case considered is a typical section at the Straight Creek Pilot bore in which at least two families of intersecting joints were found intersecting the tunnel. These cases are discussed individually below.

## 3.2 - MRDL Jointed Model Tests

The MRDL Jointed Block model is illustrated schematically in Figure 3.1. It consists of a series of blocks, square or rectangular in cross-section grouped together to form a body intersected by sets of parallel joints normal to one plane. In Figure 3.1a a typical square section block is illustrated. The square blocks are grouped as shown in Figure 3.1b with triangular blocks where required in order to form a larger mass which is square in cross-section. The model is loaded in the horizontal plane as indicated schematically in Figure 3.1c. Details of the apparatus construction, development and operation are given by Rosenblad (1971). The individual blocks are fabricated by molding using a model material consisting of sand, gypsum cement and water vibrated in a mold. The development of the model material resulted from an extensive experimentation program conducted by Rosenblad (1971) for this purpose.

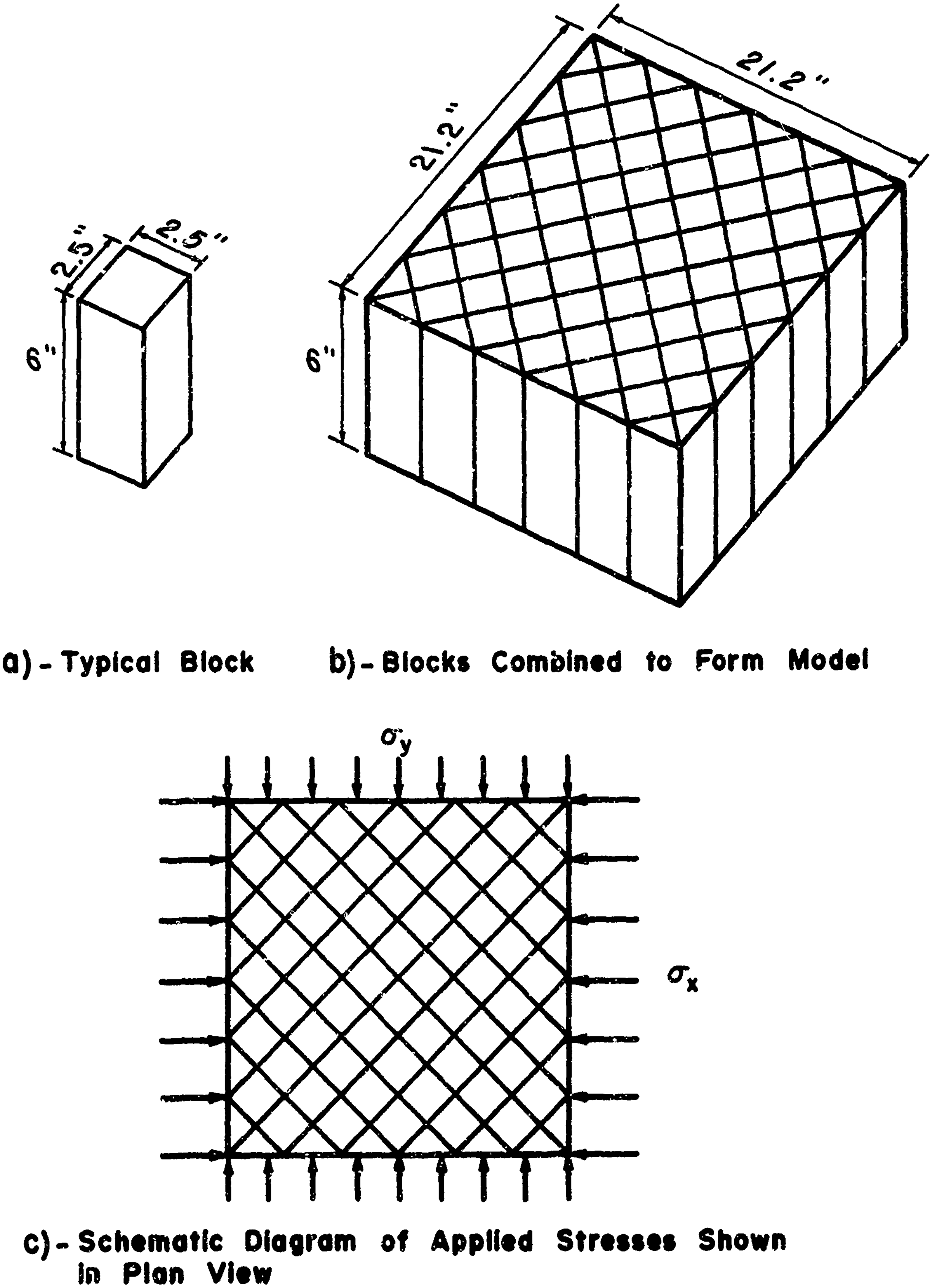



**Figure 3.1 - Schematic View of MRDL Jointed Block Model**

A typical uniaxial stress-strain curve for an intact cylindrical specimen of the vibrated model material is shown in Figure 3.2. The axial strain data were obtained from strain gages mounted in the central portion of the test sections. Rosenblad reports significant differences between the relation obtained from such strain gage measurements and those from gross measurements from the specimen. He attributes this discrepancy to end restraint effects. The importance of such effects in interpreting test results has been investigated in an earlier report (Perloff, 1969), and by Perloff and Pombo (1969). The Mohr envelopes for peak points on the stress-strain curves for the intact model material, obtained from both direct shear and triaxial compression tests, are shown in Figure 3.3.

The effect of a joint oriented at $45^{\circ}$ to the axis of a triaxial compression specimen on the stress-deformation behavior of the model material is shown in Figure 3.2. This curve is for a triaxial compression test in which the confining pressure was 500 psi. However, the equivalent Young's modulus at 50 percent peak strength was of similar magnitude for lower confining pressures. Analysis of a single-jointed specimen indicated that the results in Figure 3.2 corresponded to a joint modulus of $1.8 \times 10^{5}$ psi.

Mohr envelopes for the joints between the blocks as obtained from triaxial compression and direct shear tests are shown in Figure 3.4. As might be expected, the joints exhibit no cohesive components of shearing resistance, and have a bilinear failure envelope.

## Analysis performed

The analysis was carried out for a two-dimensional jointed model in which the blocks were assumed square. The finite element mesh used is shown in Figure 3.5. The mesh corresponds to one-quarter of the model and

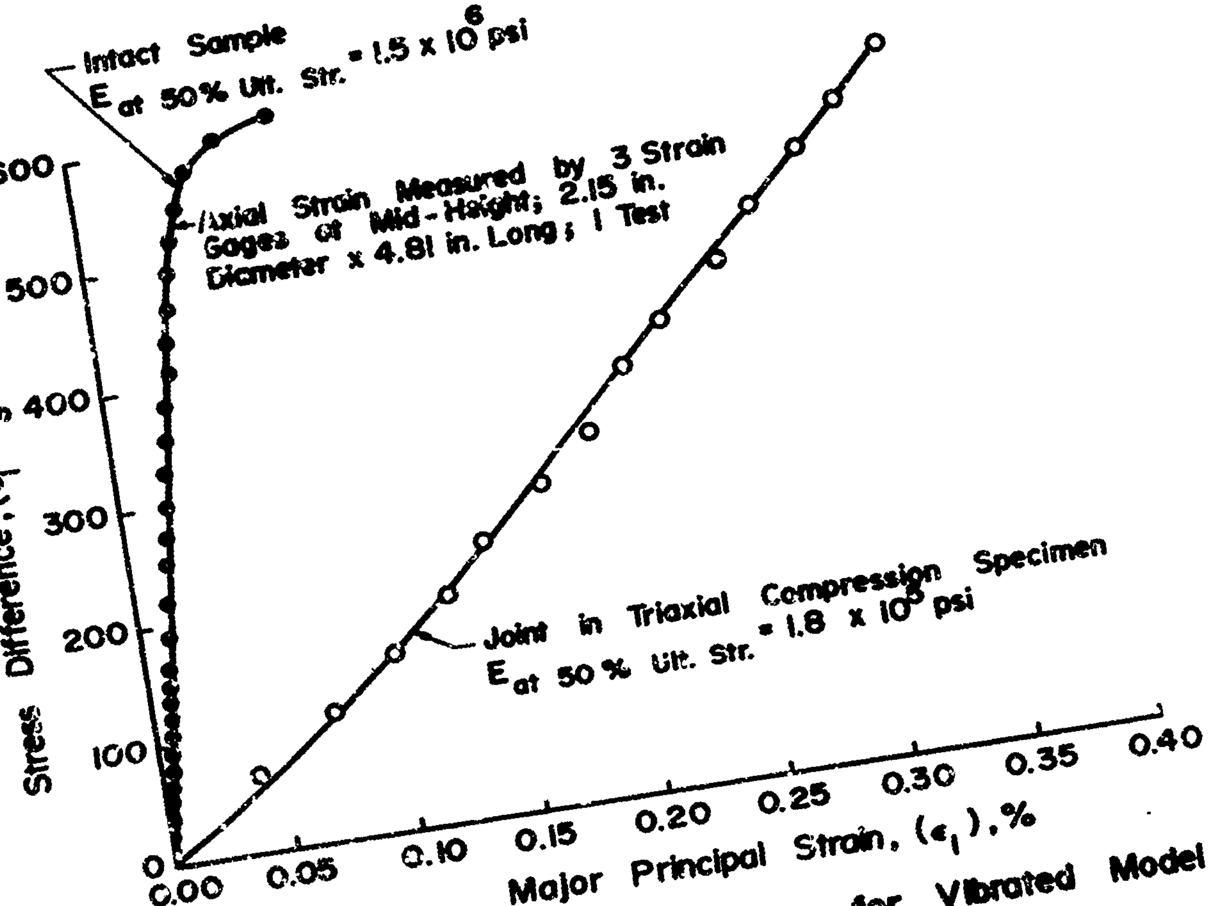



Figure 3.2 - Uniaxial Stress - Strain Relation for Vibrated Model Material

(from Rosenblad, 1971)

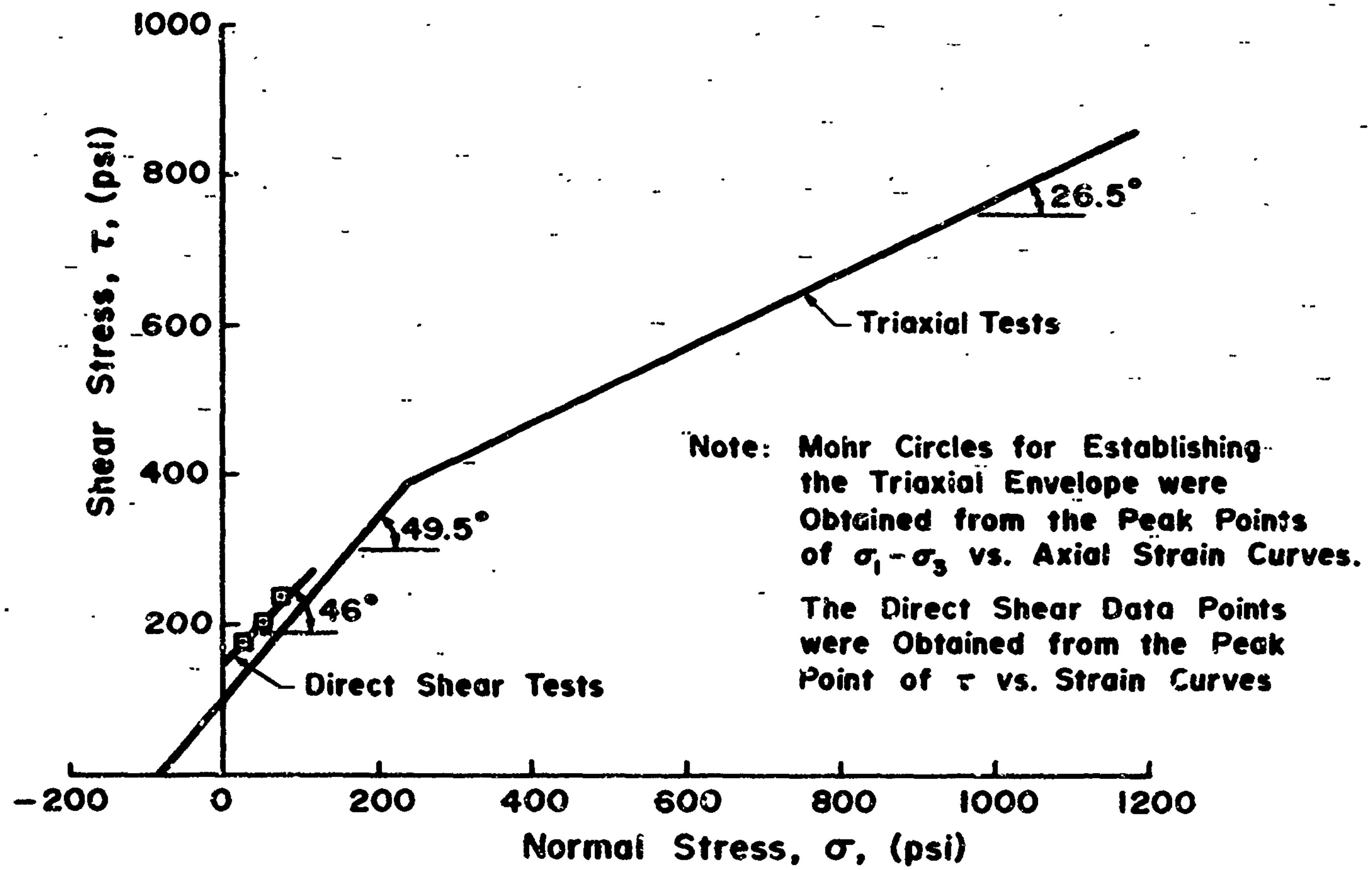



**Figure 3.3 - Mohr Envelopes for Ultimate Failure of Intact Vibrated Model Material Specimens from Triaxial and Direct Shear Tests**
(from Rosenblad, 1971)

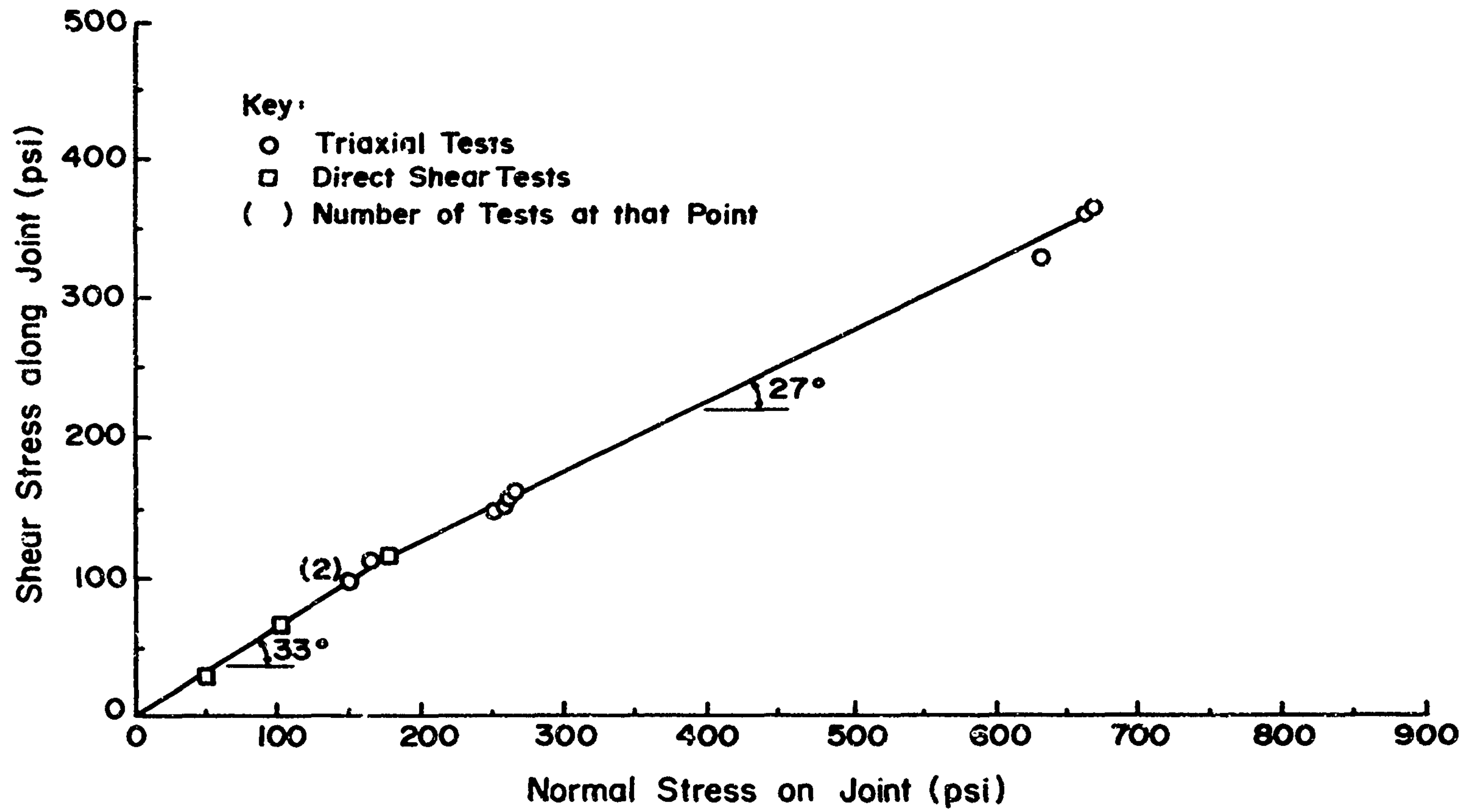



**Figure 3.4 - Mohr Envelope for Initial Slip along Joints from Triaxial and Direct Shear Tests**

(from Rosenblad, 1971)

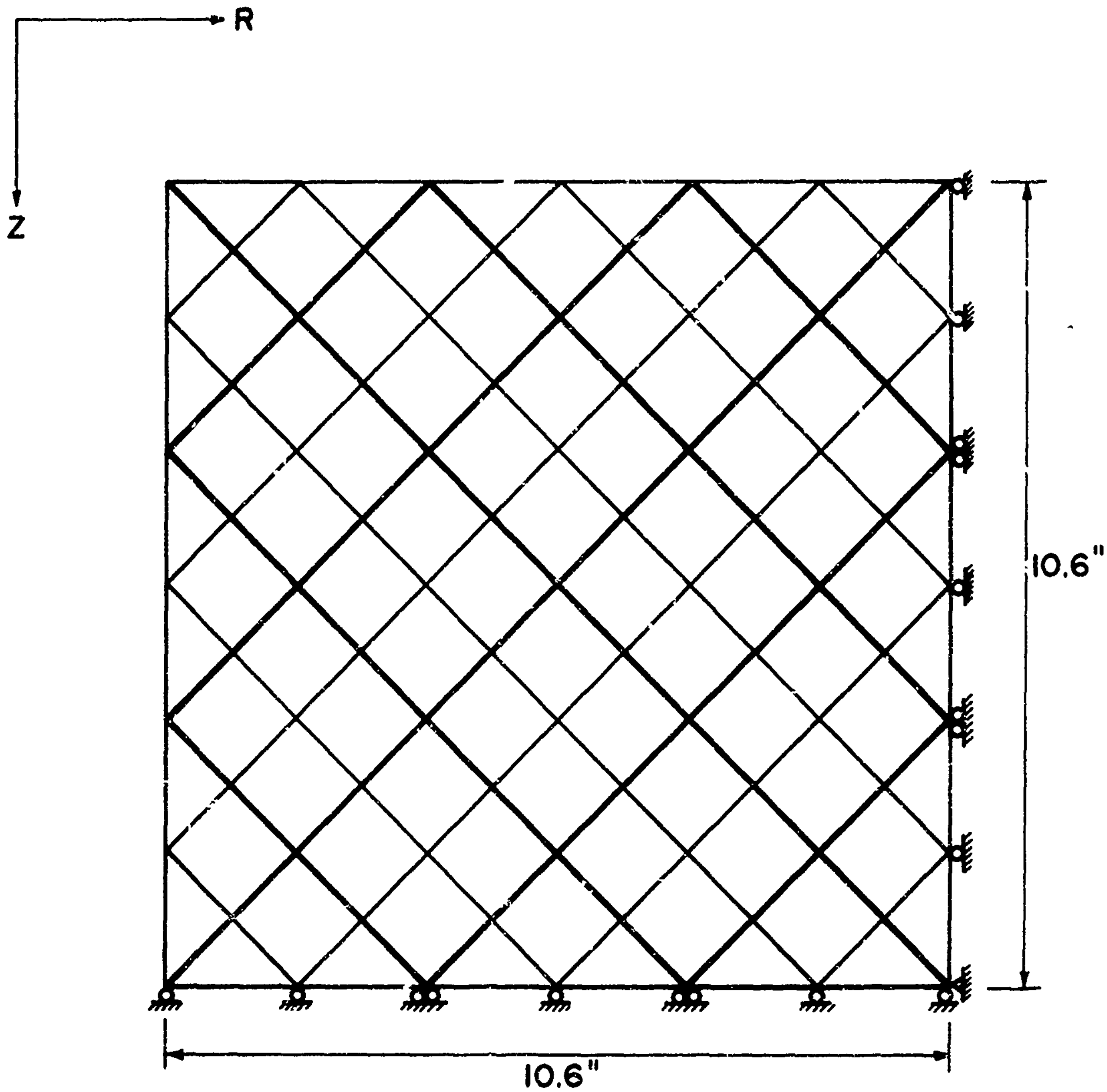



Figure 3.5 - Finite Element Mesh for MRDL Jointed Model

consists of 180 nodes and 156 elements. Each of the heavy lines indicates a joint element between intact blocks. The light lines are boundaries of elements forming the intact blocks. The material parameters used in the analysis were obtained from the test data presented by Rosenblad (1971) and shown in Figures 3.2-3.4:

Intact Blocks - Elastic-plastic material obeying the three-dimensional generalized Mohr-Coulomb yield criterion:

$E = 1.5 \times 10^6$ psi

$\nu = 0.230$

$c = 100$ psi

$\phi = 49.5$ psi ($\sigma_f \leq 230$ psi)

Joints - Elastic-plastic "material" obeying the two-dimensional bilinear Mohr-Coulomb yield criterion. Post-yield behavior is governed by residual strength parameters which are the same as those producing initial yield:

$E = 1.8 \times 10^5$ psi

$\nu = 0.230$

$c = 0$

$\phi_1 = 33^\circ$ ($\sigma_f \leq 175$ psi)

$\phi_2 = 27^\circ$ ($\sigma_f > 175$ psi)

The imposed loading used in the analysis was $\sigma_y = \sigma_2 = 25$ psi, $\sigma_z = \sigma_1$ increasing to a maximum of 275 psi.

## Results

Results of an analysis of the jointed rock model are shown as the dashed line in Figure 3.6. The line shows the relative displacement on either side of a joint, parallel to the joint, as a function of the major principal

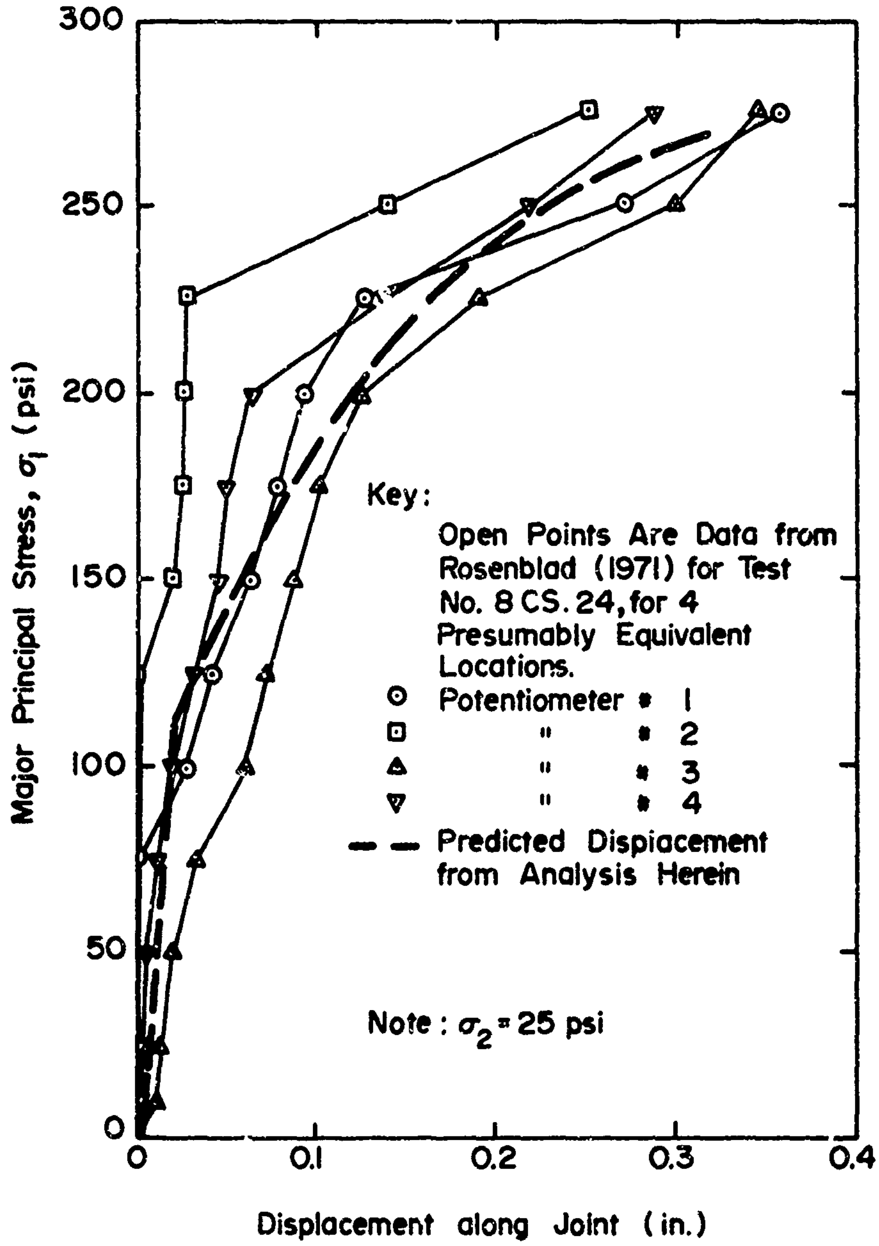



**Figure 3.6 - Comparison Between Displacement along Joints Measured in MRDL Model Study and that Predicted by Finite Element Analysis.**

stress. The curve shown corresponds to any joint because of the symmetry of the test.

Measured results for four presumably equivalent joints are also shown in Figure 3.6. The scatter in the experimental results probably arises from rotation of individual blocks, and consequent nonuniform distribution of frictional forces between the blocks, due to minor eccentricity in the jack loading system. Nonetheless, the ability of the analysis to predict the observed displacements is evident.

On the basis of these results it was concluded that the SLAM code was capable of describing the behavior of jointed systems which satisfied the following criteria:

1. The geometric arrangement of joints and intact elements can be completely described in terms of a two-dimensional system.
2. The mechanical behavior of the intact materials and individual joints can be characterized by one of the constitutive relations incorporated in the material catalogue of the code.
3. The imposed loads and displacements are known.

## 3.3 - Analysis of Straight Creek Pilot Bore

### Description of the tunnel

The Straight Creek Pilot Bore is located about 55 miles west of Denver on the proposed highway I-70. About 75 percent of the rock in the pilot bore is fine to medium grained granite (Brown, 1970). The remainder of the rock consists of metasediments that include a variety of materials. The tunnel is transected by the loveland pass fault zone, which contains numerous shear zones of diverse orientations. The rock is jointed, and joint surfaces are

commonly coated with chlorite and/or calcite.

The section chosen for analysis, Sta. 114+53, was composed of predominantly granitic rock with two major joint systems oriented approximately 38 and 52 degrees from the horizontal on a plane normal to the tunnel axis. The joint spacing observed at the tunnel wall averaged one to three feet, but was quite variable. At this location the tunnel is 250 feet below the ground surface.

Mechanical characteristics of the rock were determined by Robinson and Lee (1965). Their test results, illustrated by the Mohr circles and solid failure envelope in Figure 3.7, indicate that the intact rock obeys a Mohr-Coulomb failure criterion. Tests on samples with chlorite and calcite filled joints, in which the failure occurred along the joints, indicate peak strength behavior of the joints shown as the dashed line in Figure 3.7. Once the initial failure takes place however, it seems reasonable that the cohesive resistance diminishes to zero.

Analysis performed

The two-dimensional jointed finite element mesh used to represent the problem is shown in Figure 3.8. The heavy lines are joint elements, light lines indicate boundaries of intact elements. In the vicinity of the tunnel, joint spacing was three feet. The spacing was increased at increasing distance from the tunnel, Figure 3.8, so that there were less than 1600 modes. As in the case of the continuum analysis (Perloff, 1969), the problem is solved in two stages. The displacements resulting from the tunnel construction were determined as the difference between the displacements of the mass without the tunnel, and those with the tunnel (shaded in Figures 3.8) removed. The mesh without the tunnel consisted of 1507 nodes and 1385

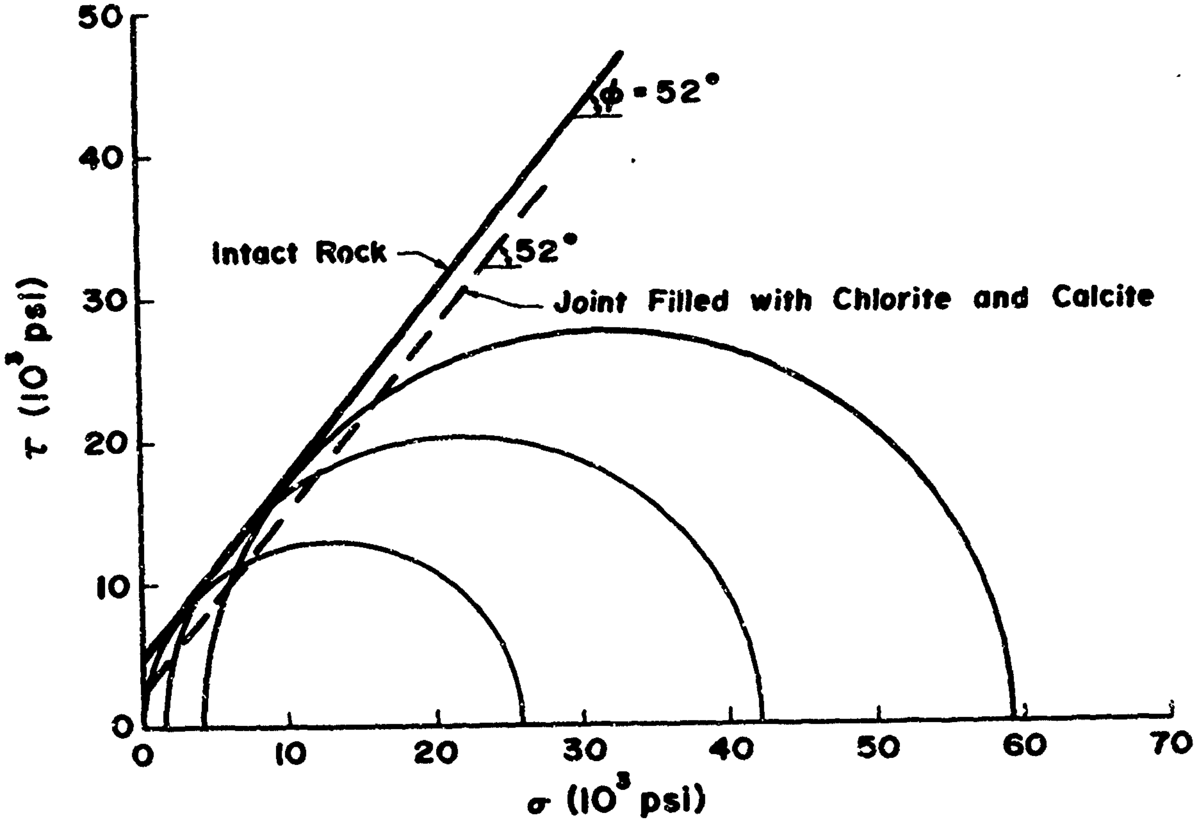



Figure 3.7 - Average Failure Envelope for Granite and Metamorphic Rocks at Straight Creek

(Data from Robinson and Lee, 1965)

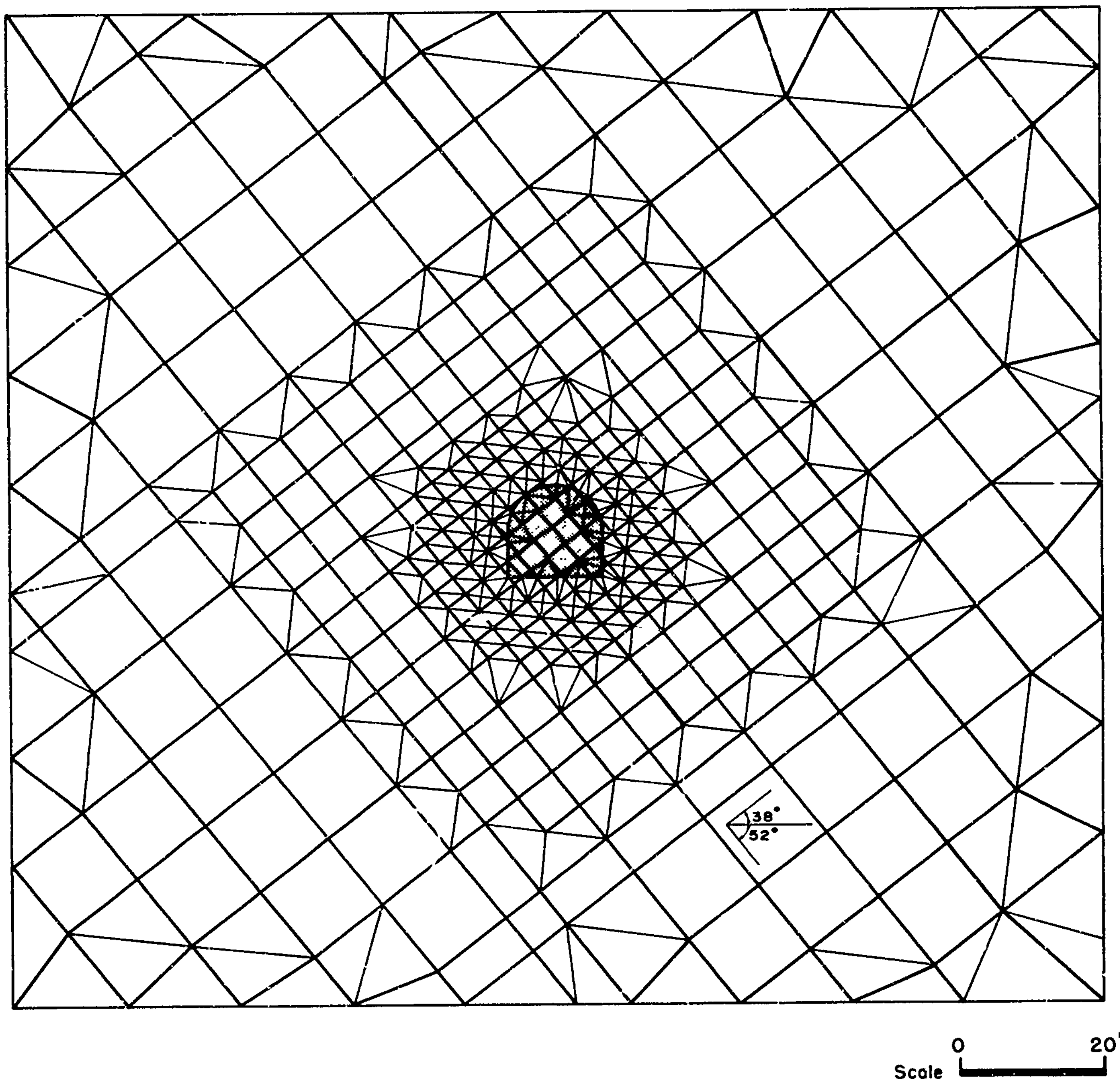



Figure 3.8 - Finite Element Mesh for Straight Creek Pilot Bore With Two Joint Systems

elements; that with the tunnel contained 1444 nodes and 1300 elements.

The material parameters used in the analysis were:

Intact rock - Elastic-plastic material obeying the three-dimensional generalized Mohr-Coulomb yield criterion:

$E = 8.98 \times 10^6$ psi

$\nu = 0.243$

$c = 4500$ psi

$\phi = 52^\circ$

Joints - Elastic-plastic material obeying the two-dimensional Mohr-Coulomb yield criterion. Post-yield behavior is cohesionless in nature as discussed above:

$E = 8.98 \times 10^5$ psi

$\nu = 0.243$

$c = 2500$ psi

$\phi_1\ \phi_2 = 52^\circ$

$c_{res} = 0$

$\phi_{res} = 52^\circ$

Imposed loads were due solely to gravity. That is, the material weight acted on all elements shown. In addition, a uniform vertical loading of 208 psi was applied to the upper boundary of the mesh to account for the overburden above the mesh.

Results

Results of the analysis and their relation to field measurements are shown graphically in Figure 3.9. This figure indicates the cross-section of the tunnel and the three MPBX's located at Sta. 114 + 53.

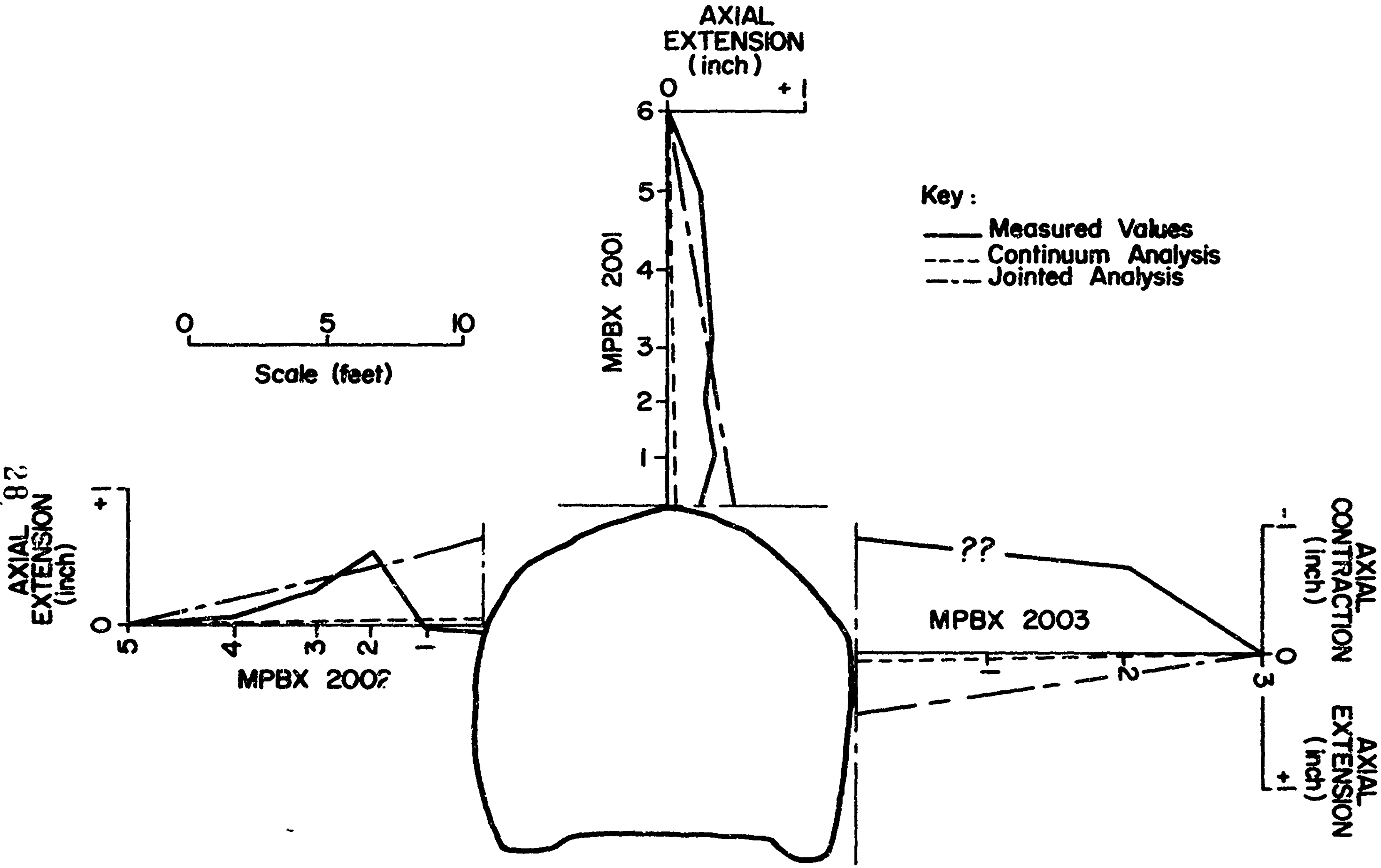



Figure 3.9 - Comparison Between Measured and Predicted Displacement at Sta 114+53 at Straight Creek Pilot Bore.

Superimposed on the diagrams of the MPBX's are the measured displacements along the MPBX axis, values calculated by the analysis for jointed systems given herein and, for comparison, the results from the continuum elastic analysis (Perloff, 1969). The analtyical results for the jointed model are closer to the measured values at MPBX's 2001 and 2002 than those for the continuum model. The calculations predict a movement in MPBX 2003 which opposite in direction to that measured. Furthermore the irregular movements recorded for MPBX 2002 near the tunnel face are not predicted by the analysis. Several possible reasons for these discrepancies can be identified. Among these are:

1. Incorrect measurement of rock movement based on MPBX data. This could arise from at least two sources:
    a. Displacements are likely to occur immediately upon excavation. Because the MPBX is installed only after excavation, important components of displacement, not necessarily in the same direction as subsequently measured values may be lost.
    b. Anchor slip may occur leading to spurious relative displacement values between individual anchors. If the anchor most remote from the tunnel slips, the whole displacement axis is translated.
2. The specific orientation and spacing of the joints in the vicinity of the MPBX's is not known but only estimated for simplified representation. This can affect predicted displacements markedly in the region near the tunnel where stress relief is the greatest.

3. The mechanical characteristics of the joints may be incorrectly described for at least two reasons:

   a. The elastic parameters for the joint materials were estimated on the basis of the measured relative moduli in the MRDL tests. Actual data on this point were not available.

   b. The joints were assumed to be unfailed, i. e., peak strength parameters were initially applicable. If the excavation process produces temporary joint separation, for example during blasting, then residual strength parameters may be more applicable.

4. Initial stress conditions are important both to the magnitude of elastic deformations as well as to the onset of yielding as the tunnel material is removed. This important point was discussed in an earlier report (Perloff, 1969) in more detail. Unfortunately, no proven means for reliably measuring the initial stress state (prior to excavation) is available at the present time.

5. The three-dimensional vature of the problem, especially the jointing and faulting undoubtedly has some influence. This point is discussed further below.

On the basis of these observations it was concluded that a more accurate prediction of tunnel behavior based on presently available input information would be fortuitous.

## SECTION 4 - CONSIDERATION OF THREE-DIMENSIONAL EFFECTS

Three-dimensional effects are undoubtedly important in the response of a rock mass to the opening of an excavation within the mass. These effects arise from at least three sources:

1. The three-dimensional nature of the tunnel geometry produces a corresponding set of stresses and deformations in the rock medium. Even in the case of a long tunnel, the geometry is decidedly three-dimensional at the end of the tunnel where construction is occurring, as well as in the vicinity of the portals.
2. The preexisting stratigraphic features, especially joints and faults, are likely to interact with the tunnel in a fashion which requires a three-dimensional framework for a realistic representation.
3. The failure criterion appropriate to the materials involved most probably involves the complete stress (or strain) field at a point. Therefore the response characteristics of the materials themselves are three-dimensional in nature.

In spite of these features, however, it was found to be impractical to incorporate three-dimensional effects into the jointed system analysis at the present time. Reasons for this included:

1. Although three-dimensional effects are likely to be significant, there are many situations (including, probably, the Straight Creek Pilot Bore) where the

other discrepancies between the real conditions and those identified are much more important. It is likely that the most important of these are the joint spacing and mechanical characteristics. Thus, only a marginal gain in accuracy seemed likely as the result of incorporating these effects at this time.

2. Even a simplified representation of the major joint systems in a two-dimensional framework required approximately 1500 nodes and 1400 elements. The SLAM code capable of managing this size problem uses approximately 53,000 words of central memory storage and approximately 15 minutes of computation time on the CDC 6500 computer. Expanding the program to three-dimensions without imposing some symmetry requirements would lead to an inordinately large problem (see for example, Corum and Krishnamurthy, 1969).

Consequently it was concluded that three-dimensional considerations could not be profitably incorporated into the analysis at the present time.

## SECTION 5 - CONCLUSIONS

Based on the results described in this report for analyses of jointed systems and measurement of performance of those systems, the following conclusions have been drawn:

1. The static SLAM code for plane jointed systems described herein can predict the response of such systems to imposed loadings when the following conditions are satisfied:
   a. The geometric arrangement of joints and intact elements can be completely described in terms of a two dimensional system.
   b. The mechanical behavior of the intact materials and individual joints can be characterized by one of the constitutive relations incorporated in the material catalogue of the code.
   c. The imposed loads and displacements are known.
2. Predictions of displacements of excavations in a natural jointed rock mass are likely to differ from measured values. The discrepancy may arise from numerous sources including:
   a. Errors inherent in the measurements themselves.
   b. Insufficient knowledge of the spacing and orientation of the joints and faults in the zone of interest.
   c. Incorrect assessment of the mechanical characteristics of the joints or joint-filling materials.

d. Inadequate knowledge of insitu stress conditions prior to excavation.

e. Three dimensional effects which cannot be incorporated in a plane analysis. This effect is likely to be less important than others mentioned above.

3. Incorporation of a three-dimensional finite-element representation of jointed systems into the analysis is not practicable at this time due both to the consequent requirement for computer storage and time, and to the limited benefit likely to be gained from such an undertaking.

## REFERENCES

Brown, L. A., (1969) "Data Bank: Underground Openings Rock Deformation Measurement-Abstracts", Technical Report No. 3, Purdue University School of Civil Engineering Project on "Strain Distribution Around Underground Openings".

Corum, J. M. and Krishnamurthy, N., (1969) "A Three-Dimensional Finite Element Analysis of a Prestressed Concrete Reactor Vessel Model", Application of Finite Element Methods in Civil Engineering, ASCE, New York, p. 63.

Costantino, C. J., (1966), "Automated Design of Advanced Hardened Facilities Study", IIT Research Institute, BSB TR-66-256.

Drucker, D. C. and Prager, W. (1952), "Soil Mechanics and Plastic Analysis or Limited Design", Quarterly of Applied Mathematics, Vol. 10, No. 2, July 1952.

Goodman, R. E., (1969) "The Deformability of Joints", Determination of the Insitu Modulus of Deformation of Rock, ASTM STP 477.

Judd, W. R. and Perloff, W. H., (1971), "Comparison between Predicted and Measured Displacements", Technical Report No. 5, Purdue University School of Civil Engineering Project on "Strain Distribution Around Underground Openings".

Obert, L., "Determination of Stress in Rock - A State of the Art Report", ASTM STP 429, ASTM, 1967.

Patton, F. D., (1966), "Multiple Mode of Shear Failure in Rock", Proceedings, First Congress of International Society of Rock Mechanics, V. 1, p. 509.

Robinson, C. S. and Lee, F. T., (1965), "Preliminary Report on the Engineering Geology of the Straight Creek Tunnel Pilot Bore, Clear Creek and Summit Counties, Colorado", Prepared in Co-operation with the Colorado Department of Highways.

Rosenblad, J. L., (1971), "Geomechanical Model Study of the Failure Modes of Jointed Rock Masses", Technical Report MRD 1-171, Missouri River Division USCE, Omaha.

Wachowski, A. and Costantino, C. J., (1966), "Automated Design of Advanced Hardened Facilities Study", BSD TR-66-256, Vol. II, Norton AFB, Calif.

## APPENDIX A - STATIC SLAM CODE FOR JOINTED SYSTEMS

### A.1 - Code Description

The Static SLAM Code is written entirely in FORTRAN IV and makes use of the overlay features of FORTRAN IV to optimize usage of the high speed core. The code consists of a main program, twelve overlays and 27 separate subroutines. The overlay structure and subroutines are shown schematically in Figure A.1.

The code uses 11 tape drives for immediate storage of data and output. The logical numbers for these tapes are 1, 2, 3, 4, 8, 9, 10, 11, 12, 14, 15. In addition, I/O is handled by tape 5 for input and 6 output. The solution is stored on tapes 3 or 12, and 15. The code is presently operational on the CDC 6500, using the Purdue MACE operating system.

### A.2 - Data Deck Setup

The following description of the data deck setup assumes that, in general, all numbers are right-oriented in their fields. Inclusion of the decimal point in floating point (real) numbers overrides the right-orientation requirement. Generally all integer data are entered in 5-column fields while all floating point data are entered in 10-column fields.

Data entered in card groups 1.1 to 6.2 are read in overlay LNK1A. Data entered in group 7.1 to 9.3 are read in overlay LNK1G. The remaining data are read in overlay LNK2.

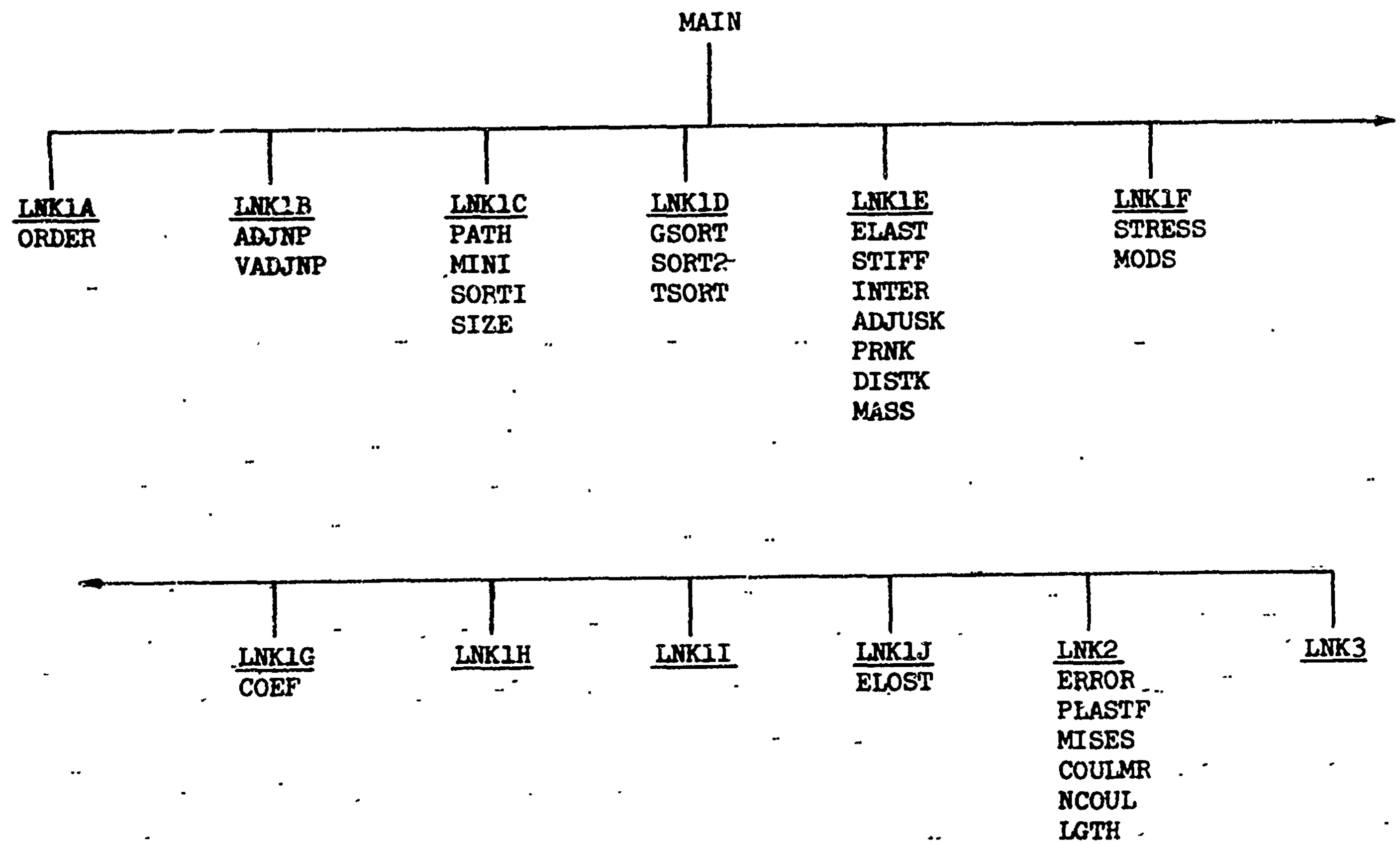



Figure A.1 - Overlay Structure of SLAM Code

| CARD | VARIABLE | FORMAT |
|---|---|---|
| 1.1 | ANAME | (18A4) |

ANAME = Problem Descriptor to be printed as output, up to 72 characters

| | | |
|---|---|---|
| 2.1 | NUMNP, NUMEL, ISTRES, IMPBX, IPRINT | (5I5) |

NUMNP = Number of node points ($\leq$ 1600)

NUMEL = Number of elements

ISTRES = Counter to describe stress condition,

= 0, axisymmetric problem,

= 1, plane strain problem,

= 2, plane stress problem.

IMPBX = Counter for storage on magnetic tape (logical number 15) for use in interpolation code (Appendix B) for determining displacement along MPBX lines,

= 0, data are not stored on tape 15

= 1, data are stored on tape 15 for subsequent use.

IPRINT = Counter for intermediate printout,

= 0, no intermediate printout other than input data,

= 1, print adjacency table and input data,

= 2, print stiffness table and input data,

= 3, print stress table and input data,

= 4, print mass vector and input data,

= 5, print load tables and input data,

= 6, print results of elimination solution and input data,

= 7, print stresses in plastic elements and input data,

=99, print all above tables.

| CARD | VARIABLE | FORMAT |
|---|---|---|
| 3.1 | N, R, Z, ITYPE, THETA | (I5,2E10.4, I10,E10.4) |

N = Node point number,

R = Radial (horizontal) coordinate (ft), increasing to the right,

Z = Vertical coordinate (ft), increasing down,

ITYPE = Counter for support conditions,

= 0, free node,

= 1, fixed in one direction,

= 2, fixed in both directions

THETA = Angle (in degrees) of roller support measured positive clockwise from the horizontal, for ITYPE = 1 only.

Note: Card 3.1 repeated NUMNP times.

| | | |
|---|---|---|
| 4.1 | NZONES | (I5) |

NZONES = Number of different materials ( $\leq$ 20)

| | | |
|---|---|---|
| 4.2 | IZ, ANAME | (I5,I8A4) |

IZ = Material or zone number

ANAME = Material or zone descriptor to be printed as output, up to 72 characters.

| | | |
|---|---|---|
| 4.3 | IELAST, IPLAST, WGT, E1, E2, E3, E4, E5 | (2I5,E10.0,5E10.0) |

IELAST = Type of linear material behavior,

= 1, isotropic elastic material,

= 2, transversely anisotropic elastic material,

= 3, linear compressible fluid.

IPLAST = Counter to describe nonlinear behavior,

= 0, elastic or linear material,

= 1, Mises (Prandtl-Reuss) elastic-plastic material,

= 2, Elastic-Plastic material with generalized three-dimensional Mohr-Coulomb yield criterion.

| CARD | VARIABLE | FORMAT |
|---|---|---|

= 3, Elastic-Plastic material with two-dimensional Mohr-Coulomb yield criterion. Applicable for joint elements only.

Note: if IPIAST = 1, 2 or 3, IELAST must equal 1 (isotropic elasticity)

WGT = Unit weight of material (pcf)

E1 to E5 = Elastic Property data.

If IELAST = 1,

E1 = Young's Modulus (psi),

E2 = Poisson's Ratio

E3 to E5 are neglected.

If IELAST = 2,

E1 = $E_r$ (psi),

E2 = $E_z$ (psi),

E3 = $E_{rz}$ (psi),

E4 = G (psi),

E5 = $E_{r\theta}$ (psi).

If IELAST = 3,

E1 = bulk modulus, and

E2 to E5 are neglected.

| | | |
|---|---|---|
| 4.4 | NOYILD | (I5) |

NOYILD = Number of nonlinear segments of effective stress-strain curve of elastic-Mises plastic material ( $\leq$ 10)

| | | |
|---|---|---|
| 4.5 | (SSTAR(I), I = 1, NOYILD) | (7E10.4) |

SSTAR = Stress (psi) at beginning of non-linear segment, up to 7 per card.

| | | |
|---|---|---|
| 4.6 | (HSTAR(I), I = 1, NOYIELD) | (7E10.4) |

HSTAR = Slope (psi) of nonlinear segment, up to 7 per card.

Note: Cards 4.4, 4.5, and 4.6 omitted if IPLAST $\neq$ 1.

| CARD | VARIABLE | FORMAT |
|---|---|---|
| 4.7 | COHESN, FRCTAN | (2E10.4) |

COHESN = Value of cohesion (psi) for generalized three-dimensional Mohr-Coulomb material, as determined from standard triaxial compression test.

FRCTAN = Corresponding friction angle (in degrees)

Note: Card 4.7 omitted if IPLAST not equal to 2.

| | | |
|---|---|---|
| 4.8 | COHESN, FRCTN1, FRCTN2, SNSWCH | (4E10.4) |

COHESN = Value of peak cohesion (psi) for two-dimensional Mohr-Coulomb material (applicable for joint elements only).

FRCTN1 = Peak friction angle (in degrees) for normal stress ≤ SNSWCH.

FRCTN2 = Peak friction angle (in degrees) for normal stress greater than SNSWCH.

SNSWCH = Normal pressure (psi) on joint at which slope angle of bilinear peak yield envelope changes from FRCTN1 to FRCTN2.

| | | |
|---|---|---|
| 4.9 | JTENSN, IRESID | (2I5) |

JTENSN = Counter indicating tensile resistance across joint

= 0, Joint material can withstand no tension normal to joint,

= 1, Joint material can resist tension normal to joint up to magnitude C/tan(FRCTN1).

IRESID = Counter indicating whether residual shear strength along joint is less than peak value.

= 0, Residual shear strength along joint = the peak value,

= 1, Residual shear strength along joint is less than peak value

| | | |
|---|---|---|
| 4.10 | CRESID, FRESID | (2E10.4) |

CRESID = Residual (post-peak) cohesion (psi)

FRESID = Residual (post-peak) friction angle (in degrees), ≤ FRCTN2.

Note: If IRESID = 0, card 4.10 is omitted.

Note: Card group 4.2 to 4.10 repeated NZONES times.

| CARD | VARIABLE | FORMAT |
|---|---|---|
| 5.1 | NUME, IZONE, NPI, NPJ, NPK, NPL, NCRACK | (7I5) |

NUME = Element number,

IZONE = Material zone number in which element is located,

NPI to NPL = Node numbers comprising element. If NPL = 0, element is considered triangle. If element is a joint element, NPI must be either node with smallest R-coordinate. If the element is vertical, NPI must be either node with the smallest z-coordinate. NPJ, NPK, NPL must be nodes given in clockwise order around the element. For all other element types, there is no restriction on ordering of nodes.

NCRACK = Counter to identify joint elements,

= 0, regular triangular or rectangular element,

= 1, rectangular joint element of zero thickness. Thus two nodes will have the same coordinate, and the other two nodes will have the same coordinates.

Note: Card 5.1 repeated NUMEL times.

| | | |
|---|---|---|
| 6.1 | NUMST | (I5) |

NUMST = Number of start nodes for renumbering scheme (< 80)

| | | |
|---|---|---|
| 6.2 | (NSTART(I), I = 1, NUMST) | (14I5) |

NSTART = Start node numbers, 14 per card.

| | | |
|---|---|---|
| 7.1 | NLINES | (I5) |

NLINES = Number of surfaces along which applied pressure acts.

| | | |
|---|---|---|
| 7.2 | LOADNP, ANAME | (I5,18A4) |

LOADNP = Number of node points that are loaded by pressure on one surface (≤ 100).

ANAME = Pressure descriptor to be printed as output, up to 72 characters.

| | | |
|---|---|---|
| 7.3 | NPLOAD, PRESSU, PRESSW | (I5,2E10.0) |

NPLOAD = Node number of node to which pressure is applied.

PRESSU = Horizontal pressure (psi) applied to loaded surface at node number NPLOAD, positive in positive R-direction (to the right).

| CARD | VARIABLE | FORMAT |
|---|---|---|

PRESSW = Vertical pressure (psi) applied to loaded surface at node number NPLOAD, positive in positive z-direction (down).

Note: Card 7.3 repeated LOADNP times. Loaded node numbers, NPLOAD, are in consecutive order along pressure surface such that in moving from the first to the last the pressures are applied on the left hand side of the surface, and the boundary element is to the right of the surface.

Note: Card group 7.2 through 7.3 repeated NLINES times. If NLINES = 0, they are omitted.

| | | |
|---|---|---|
| 8.1 | NLINES | (I5) |

NLINES = Number of clusters of nodes to which concentrated loads are applied.

| | | |
|---|---|---|
| 8.2 | LOADNO, ANAME | (I5,18A4) |

LOADNP = Number of nodes in cluster (≤ 100)

ANAME = Concentrated load cluster descriptor to be printed as output, up to 72 characters.

| | | |
|---|---|---|
| 8.3 | NPLOAD, PLOADU, PLOADW | (I5,2E10.0) |

NPLOAD = Node number of node to which pressure is applied.

PLOADU = Horizontal force (lbs.) applied to node number NPLOAD, positive in positive R-direction (to the right).

PLOADW = Vertical force (lbs.) applied to node number NPLOAD, positive in positive Z-direction (down).

Note: Card 8.3 repeated LOADNP times.

Note: Card group 8.2 through 8.3 repeated NLINES times. If NLINES = 0, they are omitted.

| | | |
|---|---|---|
| 9.1 | NLINES | (I5) |

NLINES = Number of clusters of nodes for which displacements are specified.

| | | |
|---|---|---|
| 9.2 | LDISP, ANAME | (I5,18A4) |

LDISP = Number of nodes in displacement cluster.

ANAME = Displacement cluster descriptor to be printed as output, up to 72 characters.

| CARD | VARIABLE | FORMAT |
| --- | --- | --- |
| 9.3 | NPDISP, UDISP, WDISP | (I5,2E10.0) |

NPDISP = Node number of node for which displacements are specified.

UDISP = u - displacement (inches)[1]

WDISP = w - displacement (inches)[1]

Note: Card 9.3 is repeated LDISP times.

Note: Card group 9.2 through 9.3 repeated NLINES times. If NLINES = 0, they are omitted.

| | | |
| --- | --- | --- |
| 10.1 | ITMAX, ERRMAX, NFAC, KTAPE, ICONTU, OVERLX | (I5,E10.0,3I5,E10.0) |

ITMAX = Maximum number of iterations to be used for each solution increment.

ERRMAX = Maximum allowable error (lbs) in force computations.

NFAC = Number of increments to be used in nonlinear solution to go from loads at which stresses reach the elastic limit to the actual loads.

KTAPE = Counter for use of tapes for storage of node point data,

= 0, uses two K tapes (normal usage),

= 1, uses only 1 K tape.

ICONTU = 0 will continue complete solution if ITMAX is reached in any load increment without convergence to within the allowable error, (ERRMAX).

OVERLX = Over-relaxation factor to reduce required number of iterations (usual values 1.2 - 1.8).

---

1. When ITYPE = 1 on card 3.1, only one displacement component is specified at the particular node point. This is equivalent to a roller support, as shown in Figure A.2, in which the roller is free to move along a line oriented at the angle θ (also input on card 3.1) with respect to the horizontal. The angle θ is considered positive when measured clockwise from the horizontal, and defines a new set of coordinates u, w as shown in Figure A.2. The direction in which the roller is free to move is defined as the $\bar{u}$ direction. The direction in which the displacement is specified is defined as the $\bar{w}$ direction. Thus in the case of the equivalent roller support node point, WDISP is the specified displacement in thw $\bar{w}$ direction, and UDISP is ignored. Similarly in the output, the result given for u is the displacement in the $\bar{u}$ direction and w is the specified input displacement in the $\bar{w}$ direction.

Forces at nodes with equivalent roller supports are specified in the usual horizontal and vertical coordinate directions.

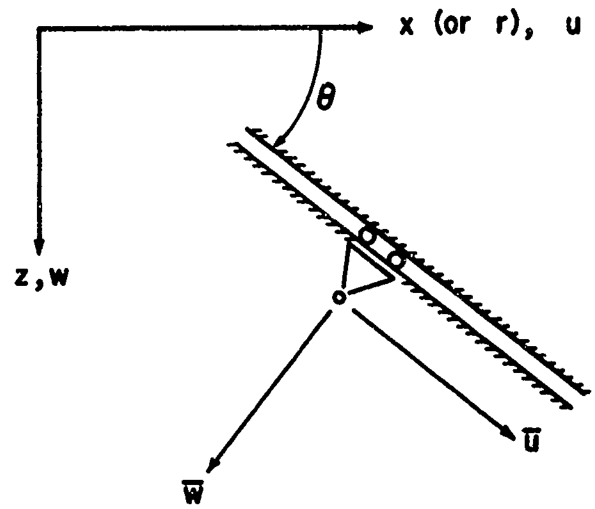



Figure A.2- Coordinate Directions for Equivalent Roller Support

A.3 - Listing of Code

```
      OVERLAY(MOHAN,0,0        )
      PROGRAM SLAM(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE1,TAPE2,TAPE3,TAPE4,TAPE8,TAPE9,TAPE10,TAPE11,TAPE12,
     2TAPE14,TAPE15,TAPE16)

      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1       NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2       KTAPE,KRUN,IPRINT,NUMST,MXSTRT,IELAST(20),IPLAST(20),
     3    WGT(20),NSTART(79),EI(5,20),IPELTP,INT,NPRCDS,IMPBX
C
C
      MXCLS=80
      MXNPB=350
      MAXNP=1600
      MXADJP=8
      MXZONE=20
      MXSTRT=79
      MXPELB=24
      KRUN=0
C
      MOHAN=5HMOHAN
C
      CALL OVERLAY(MOHAN,1,0,6HRECALL)
C
      CALL OVERLAY(MOHAN,2,0,6HRECALL)
C
      CALL OVERLAY(MOHAN,3,0,6HRECALL)
C
      CALL OVERLAY(MOHAN,4,0,6HRECALL)
C
      CALL OVERLAY(MOHAN,5,0,6HRECALL)
C
      CALL OVERLAY(MOHAN,6,0,6HRECALL)
C
      CALL OVERLAY(MOHAN,7,0,6HRECALL)
C
      CALL OVERLAY(MOHAN, 8,0,6HRECALL)
C
      CALL OVERLAY(MOHAN, 9,0,6HRECALL)
C
      CALL OVERLAY(MOHAN,10,0,6HRECALL)
C
      CALL OVERLAY(MOHAN,11,0,6HRECALL)
C
      CALL OVERLAY(MOHAN,12,0,6HRECALL)
C
      STOP
      END
```

```
C
C
C
      OVERLAY(MOHAN,1,0          )
      PROGRAM LNK1A
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1       NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2       KTAPE,KRUN,IPRINT,NUMST,MXSTRT,IELAST(20),IPLAST(20),
     3 WGT(20),NSTART(79),EI(5,20),IPELTP,INT,NPRCDS,IMPBX
C
      DIMENSION R(1600),Z(1600),ITYPE(1600),THETA(1600)
C
C
      DIMENSION ANAME(18),SSTAR(10),HSTAR(10)
C
C**** SYSTEM DATA
      MOHAN=5HMOHAN

C
      READ(5,100) ANAME,NUMNP,NUMEL,ISTRES,IMPBX,IPRINT
  100 FORMAT(18A4/14I5)
C
C     ANAME =PROBLEM TITLE
C     NUMNP =NO. OF NODE POINTS
C     NUMEL =NO. OF ELEMENTS
C     ISTRES=0  AXISYMMETRIC PROBLEM
C           =1  PLANE STRAIN PROBLEM
C           =2  PLANE STRESS PROBLEM
C     IPRINT=0  ONLY ECHO PRINT INPUT DATA
C           =1  ONLY PRINT ADJACENCY TABLE
C           =2  ONLY PRINT STIFFNESS TABLE
C           =3  ONLY PRINT STRESS TABLE
C           =4  ONLY PRINT MASS VECTOR
C           =5  ONLY PRINT LOAD TABLES
C           =6  ONLY PRINT RESULTS OF ELIMINATION
C           =7  ONLY PRINT STRESSES IN PLASTIC ELEMENTS
C           =99 PRINT ALL ABOVE TABLES
C
      WRITE(6,101) ANAME,NUMNP,NUMEL,IPRINT
  101 FORMAT(1H1,18A4//20H NO. OF NODE POINTS=,I5/
     1                 20H NO. OF ELEMENTS    =,I5/
     2                 20H IPRINT             =,I5)
      IF(ISTRES.NE.0) GO TO 103
      WRITE(6,104) ISTRES
  104 FORMAT(20H ISTRES             =,I5,3X,20HAXISYMMETRIC  PROBLEM)
      GO TO 102
  103 IF(ISTRES.NE.1) GO TO 105
      WRITE(6,106) ISTRES
  106 FORMAT(20H ISTRES             =,I5,3X,20HPLANE  STRAIN  PROBLEM)
      GO TO 102
  105 IF(ISTRES.NE.2) GO TO 107
      WRITE(6,108) ISTRES
  108 FORMAT(20H ISTRES             =,I5,3X,20HPLANE  STRESS  PROBLEM)
      GO TO 102
  107 WRITE(6,109) ISTRES
  109 FORMAT(20H ISTRES             =,I5,3X,20HERROR  IN ISTRES DATA)
      CALL EXIT
C
C**** READ NODE POINT DATA AND STORE ON TAPE 14
C
```

```
  102 PI=3.1415927

      REWIND 14
      WRITE(6,110)
  110 FORMAT(1H1,15HNODE POINT DATA//12X,7HNODE PT,4X,4HTYPE,13X,
     15HTHETA,15X,6HRADIUS,14X,5HDEPTH/14X,3HNO.,21X,9H(DEGREES),
     214X,4H(FT),16X,4H(FT) //)
C
      DO 111 I=1,NUMNP
  111 READ(5,112) NPN,R(NPN),Z(NPN),ITYPE(NPN),THETA(NPN)
  112 FORMAT(I5,2E10.4,I10,E10.4)
C
C     NPN   =NODE POINT NO.
C     R     =RADIUS (FT)
C     Z     =DEPTH  (FT)
C     ITYPE =0 FREE NODE
C           =1 FIXED NODE IN ONE DIRECTION (EXCEPT Z-AXIS)
C           =2 FIXED NODE IN TWO DIRECTIONS(INCLUDING Z-AXIS)
C           =3 FREE NODE ON Z-AXIS (NOT NECESSARY TO SPECIFY)
C     THETA =CLOCKWISE ANGLE OF ROLLER DIRECTION FROM
C            R-AXIS (DEGREES)
C***********************************************************************
      IF (IMPBX.NE.1) GO TO 15
      REWIND 15
      WRITE(15)NUMNP,(R(I),Z(I),I=1,NUMNP)
C***********************************************************************
C
  15  DO 16 NPN=1,NUMNP
      IF(R(NPN).NE.0.0) GO TO 17
      IF(ITYPE(NPN).EQ.2) GO TO 17
      IF(ISTRES.NE.0) GO TO 17
      ITYPE(NPN)=3
   17 WRITE(6,113)  NPN,ITYPE(NPN),THETA(NPN),R(NPN),Z(NPN)
  113 FORMAT(I16,I9,5X,1P3E20.7)
      R(NPN)=R(NPN)*12.0
      Z(NPN)=Z(NPN)*12.0
      THETA(NPN)=THETA(NPN)*PI/180.
      WRITE(14) NPN,R(NPN),Z(NPN),ITYPE(NPN),THETA(NPN)
   16 CONTINUE
C
C**** READ ZONE PROPERTY DATA AND STORE ON TAPE 14
C
      READ(5,114) NZONES
  114 FORMAT(14I5)
      WRITE(6,118)NZONES
  118 FORMAT(1H1,18HZONE PROPERTY DATA/14H NO. OF ZONES=,I5)
      WRITE(14) NZONES
      DO 1 I=1,NZONES
      READ(5,119) IZ,ANAME
  119 FORMAT(I5,18A4)
C
C     NZONES=NO. OF MATERIAL ZONES
C     IZ    =ZONE NUMBER
C     ANAME =ZONE DESCRIPTOR
C
      WRITE(6,120) IZ,ANAME
  120 FORMAT(//13H ZONE NUMBER=,I5,2X,18A4)
C
      READ(5,121) IELAST(IZ),IPLAST(IZ),WGT(IZ),(EI(J,IZ),J=1,5)
  121 FORMAT(2I5,F10.0,5E10.0)
```

```
C
C     WGT   =UNIT WEIGHT (LB/FT3)
C     E1    =ELASTIC MODULUS (PSI) FOR ISOTROPIC MATERIAL
C     E2    =POISSON,S RATIO FOR ISOTROPIC MATERIAL
C     E3,4,5=PARAMETERS FOR ANISOTROPIC ELASTICITY
C     IELAST=1, ISOTROPIC ELASTICITY
C           =2, ANISOTROPIC ELASTICITY
C           =3, COMPRESSIBLE FLUID
C     IPLAST=0, LINEAR MATERIAL
C           =1, VON MISES PLASTICITY
C           =2, THREE-DIMEN. MOHR-COULOMB MATERIAL
C           =3, TWO-DIMEN. MOHR-COULOMB MATERIAL, JOINT ELEMENTS ONLY
C
      WRITE(6,122) IELAST(IZ),IPLAST(IZ),WGT(IZ)
  122 FORMAT(10X,20HIELAST                  =,I5/
     1       10X,20HIPLAST                  =,I5/
     2       10X,20HUNIT WEIGHT             =,1PE15.5,2X,3HPCF)
C
      IF((IELAST(IZ).EQ.0).OR.(IELAST(IZ).GT.3)) GO TO 400
      IF(IELAST(IZ).GT.1) GO TO 123
      WRITE(6,124) EI(1,IZ),EI(2,IZ)
  124 FORMAT(10X,20HELASTIC MODULUS         =,1PE15.5,2X,3HPSI/
     1       10X,20HPOISSON,S RATIO         =,1PE15.5)
      GO TO 500
  123 IF(IELAST(IZ).GT.2) GO TO 125
      WRITE(6,137)  EI(1,IZ),EI(2,IZ),EI(3,IZ),EI(4,IZ),EI(5,IZ)
  137 FORMAT(10X,20HE1                      =,1PE15.5/
     1       10X,20HE2                      =,1PE15.5/
     2       10X,20HE3                      =,1PE15.5/
     3       10X,20HE4                      =,1PE15.5/
     4       10X,20HE5                      =,1PE15.5)
      GO TO 500
  125 WRITE(6,126) EI(1,IZ)
  126 FORMAT(10X,20HBULK MODULUS            =,1PE15.5,2X,3HPSI)
      GO TO 500
C
  400 WRITE(6,401)
  401 FORMAT(19H ERROR IN ZONE DATA)
      CALL EXIT
C
  500 WRITE(14) IZ,IELAST(IZ),IPLAST(IZ),WGT(IZ),(EI(J,IZ),J=1,5)
C
      IF(IPLAST(IZ).GT.3) GO TO 400
      IF(IPLAST(IZ).EQ.0) GO TO 1
      IF(IPLAST(IZ).GT.1) GO TO 200
C
C     MISES MATERIAL DATA
C
      IF(IELAST(IZ).NE.1) GO TO 400
      READ(5,114) NOYILD
      READ(5,127) (SSTAR(J),J=1,NOYILD)
      READ(5,127) (HSTAR(J),J=1,NOYILD)
  127 FORMAT(7E10.4)
C
C     NOYILD=NO. OF NONELASTIC STRAIGHT LINE SEGMENTS ON
C            UNIAXIAL STRESS-STRAIN CURVE
C     SSTAR =STRESS AT BEGINNING OF SEGMENT (PSI)
C     HSTAR =SLOPE OF SEGMENT (PSI)
C
      WRITE(6,128) NOYILD
```

```
  128 FCRMAT(10X,20HNO. OF PL. SEGMENTS=,I5)
      WRITE(6,139) SSTAR(1)
  139 FCRMAT(10X,20HSTRESS AT START     =,1PE15.5,2X,3HPSI)
      IF(NOYILD.EQ.1) GO TO 141
      WRITE(6,140) (SSTAR(J),J=2,NOYILD)
  140 FORMAT(29X,1H=,1PE15.5,2X,3HPSI)
  141 WRITE(6,142) HSTAR(1)
  142 FCRMAT(10X,20HSLOPE OF EL. CURVE =,1PE15.5,2X,3HPSI)
      IF(NOYILD.EQ.1) GO TO 143
      WRITE(6,140) (HSTAR(J),J=2,NOYILD)
  143 DC 129 J=1,NOYILD
      IF(HSTAR(J).GE.EI(1,IZ)) GO TO 400
  129 HSTAR(J)=EI(1,IZ)*HSTAR(J)/(EI(1,IZ)-HSTAR(J))
      WRITE(6,144) HSTAR(1)
  144 FCRMAT(10X,20HSLOPE OF PL. CURVE =,1PE15.5,2X,3HPSI)
      IF(NOYILD.EQ.1) GO TO 145
      WRITE(6,140) (HSTAR(J),J=2,NOYILD)
  145 WRITE(14) NOYILD,(SSTAR(J),J=1,NOYILD),(HSTAR(J),J=1,NOYILD)
      GC TO 1
C
C     THREE-DIMENSIONAL MOHR-COULOMB MATERIAL
C
  200 IF(IPLAST(IZ).GT.2) GO TO 300
      IF(IELAST(IZ).NE.1) GO TO 400
      READ(5,127) COHESN,FRCTAN
C
C     CCHESN=SOIL COHESION (PSI) FROM TRIAXIAL TEST
C     FRCTAN=FRICTION ANGLE (DECREES)
C
      WRITE(6,201) COHESN,FRCTAN
  201 FCRMAT(10X,20HCOHESION, TRIAXIAL =,1PE15.5,2X,3HPSI/
     1          10X,20HFRICTION ANGLE      =,1PE15.5,2X,7HDEGREES)
      FRCTAN=FRCTAN*PI/180.
      ALPHA=(2./SQRT(3.))* SIN(FRCTAN)/(3.-SIN(FRCTAN))
      CAPPA=(6./SQRT(3.))*COHESN*COS(FRCTAN)/(3.-SIN(FRCTAN))
      WRITE(6,202) ALPHA,CAPPA
  202 FCRMAT(10X,20HYIELD COEF, ALPHA  =,1PE15.5/
     1          10X,20HYIELD COEF, K       =,1PE15.5,2X,3HPSI)
      COSTH=SQRT(ALPHA**2./(ALPHA**2.+(1./6.)))
      IF((ISTRES.EQ.2).AND.(CAPPA.EQ.0.0)) GO TO 400
      WRITE(14) ALPHA,CAPPA,COSTH
      GC TO 1
C
C     TWO-DIMEN. MOHR-COULOMB MATERIAL FOR JOINTS ONLY
C
 300  READ(5,127) COHESN,FRCTN1,FRCTN2,SNSWCH
C
C     COHESN=COHESIVE COMPONENT OF PEAK STRENGTH(PSI)
C     FRCTN1=INITIAL PHI ANGLE FOR BILINEAR FAILURE ENVELOPE(DEGREES)
C     FRCTN2=PHI ANGLE FOR BILINEAR FAILURE ENVELOPE WHEN NORMAL
C             STRESS GREATER THAN SNSWCH (DEGREES)
C     SNSWCH=NORMAL STRESS AT WHICH BILINEAR FAILURE
C             ENVELOPE CHANGES SLOPE
C
      WRITE(6,301) COHESN,FRCTN1,FRCTN2,SNSWCH
 301  FORMAT(10X,20HCOHESION              =,1PE15.5,2X,3HPSI/
     1          10X,20HINITIAL PHI ANGLE  =,1PE15.5,2X,7HDEGREES/
     2          10X,20HSECOND BILINEAR PHI=,1PE15.5,2X,7HDEGREES/
     3          10X,17HNORMAL STRESS FOR/
     4          12X,17HBREAK IN BILINEAR/
```

```
     5        12X,15HFAILURE ENVELOPE =,1PE15.5,3HPSI)
      SNSWCH=-SNSWCH
C
      READ(5,114) JTENSN, IRESID
C
C     JTENSN=0, JOINT MATERIAL CAN WITHSTAND NO TENSION NORMAL TO JOINT
C           =1, JOINT MATERIAL CAN WITHSTAND TENSION NORMAL TO JOINT
C               UP TO MAGNITUDE COHESN/TAN(FRCTN1)
C
C     IRESID=0, RESIDUAL SHEAR STRENGTH AFTER YEILD=PEAK SHEAR STRENGTH
C           1, RESIDUAL SHEAR STRENGTH LESS THAN PEAK
C
      IF(JTENSN.EQ.0) WRITE(6,305)
  305 FORMAT(10X,20HJOINT MATERIAL TAKES/
     1        12X,17HNO TENSION NORMAL/
     2        12X,8HTO JOINT)
C
      IF(IRESID.EQ.1) GO TO 310
      WRITE(6,307)
  307 FORMAT(10X,14HRESIDUAL SHEAR/
     1        12X,15HSTRENGTH = PEAK)
      CRESID = COHESN
      FRESID = FRCTN1
      GO TO 320
C
  310 READ (5,127) CRESID,FRESID
C
C     CRESID = RESIDUAL COHESION (PSI)
C     FRESID = RESIDUAL PHI ANGLE (DEGREES)
C
      WRITE(6,315) CRESID,FRESID
  315 FORMAT(10X,20HRESIDUAL COHESION   =,1PE15.5,2X,3HPSI/
     1        10X,20HRESIDUAL PHI ANGLE =,1PE15.5,2X,7HDEGREES)
C
C
  320 FRCTN1=FRCTN1*PI/180.
      FRCTN2=FRCTN2*PI/180.
      FRESID=FRESID*PI/180.
      MYIELD = 0
      WRITE(14)COHESN,FRCTN1,FRCTN2,SNSWCH,CRESID,
     1    FRESID,MYIELD,IRESID,JTENSN
      GO TO 1
C
    1 CONTINUE
C
C**** READ ELEMENT DATA, REORDER ELEMENT NODES, OUT ON TAPE 1
C     PLASTIC ELEMENTS ON TAPE 14
C
      REWIND 1
      NUMPEL=0
      WRITE(6,131)
  131 FORMAT(1H1,12HELEMENT DATA//12X,7HELEMENT,5X,4HZONE,6X,3HNPI,
     17X,3HNPJ,7X,3HNPK,7X,3HNPL,7X,6HNCRACK/14X,3HNO. 8X,3HNO.//)
C****************************************************************************
      IF(IMPBX.EQ.1)WRITE(15)NUMEL
C****************************************************************************
C
      DO 7 M=1,NUMEL
      NPL=0
      READ(5,114)  NUME,IZONE,NPI,NPJ,NPK,NPL,NCRACK
```

```
C
C       NUME   =ELEMENT NUMBER
C       IZONE =ZONE NUMBER
C       NP     =NODE POINT NUMBERS
C       NCRACK=0 REGULAR ELEMENT, =1 CRACK MODEL
C
        CALL ORDER(NPI,NPJ,NPK,NPL,R,Z,ISTRES,KASE,MAXNP,NCRACK)
        WRITE(1) NUME,IZONE,KASE,NPI,NPJ,NPK,NPL,NCRACK
        WRITE(6,132) NUME,IZONE,NPI,NPJ,NPK,NPL,NCRACK
  132 FORMAT(I16,I11,3X,I6,4X,I6,4X,I6,4X,I6,4X,I6)
C*************************************************************************
        IF(IMPBX.EQ.1)WRITE(15)NUME,NPI,NPJ,NPK,NPL
C*************************************************************************
C
        IF(IPLAST(IZONE).EQ.0) GO TO 7
        NUMPEL=NUMPEL+1
        IF(NPL.NE.0) GO TO 133
        ITL=0
        THL=0.0
        RL=0.0
        ZL=0.0
        GO TO 134
  133 ITL=ITYPE(NPL)
        THL=THETA(NPL)
        RL=R(NPL)
        ZL=Z(NPL)
  134 WRITE(14) NUME,IZONE,IPLAST(IZONE),NPI,NPJ,NPK,NPL,NCRACK,
     1ITYPE(NPI),ITYPE(NPJ),ITYPE(NPK),ITL,
     2THETA(NPI),THETA(NPJ),THETA(NPK),THL,
     3     R(NPI),     R(NPJ),     R(NPK),RL,
     4     Z(NPI),     Z(NPJ),     Z(NPK),ZL
C
    7 CONTINUE
C
        WRITE(6,150) NUMPEL
  150 FORMAT(1H1,26HNO. OF NONLINEAR ELEMENTS=,I5)
C
C**** STARTING NODE DATA FOR PATH ROUTINE
C
        READ(5,114) NUMST
        READ(5,114) (NSTART(I),I=1,NUMST)
C       NUMST = NO. OF STARTING NODES(LT.100)
C       NSTART=STARTING NODE NUMBERS
C
        WRITE(6,135) NUMST
  135 FORMAT(1H1,18HSTARTING NODE DATA//20H NO. OF START NODES=,I5//)
        WRITE(6,136) (NSTART(I),I=1,NUMST)
  136 FORMAT(22H STARTING NODE NUMBERS/(15X,10I7))
C
C**** AT THIS TIME, TAPE 14 HAS ORIGINAL NODE POINT DATA
C                                      ZONE DATA
C                                      PLASTIC ELEMENT DATA, ORIGINAL NODE ORDE
C                      TAPE  1 HAS ALL ORIGINAL ELEMENT DATA
C
        REWIND 14
        REWIND  1
        RETURN
        END
C
C
```

```
C
      SUBROUTINE ORDER(NPI,NPJ,NPK,NPL,R,Z,ISTRES,KASE,MAXNP,NCRACK)
      DIMENSION R(MAXNP),Z(MAXNP)
C
C**** ORDER NODE POINT LETTERING FOR ELEMENT AND DEFINE CASE
C
C     R      =RADIAL    COORDINATE OF NODE POINT
C     Z      =VERTICAL COORDINATE OF NODE POINT
C     KASE  =1, GENERAL TRIANGLE
C            =2, TRIANGLE, ONE NODE ON Z-AXIS
C            =3, TRIANGLE, TWO NODES ON Z-AXIS
C            =4, GENERAL RECTANGLE
C            =5, RECTANGLE, ONE NODE ON Z-AXIS
C            =6, RECTANGLE, TWO NODES ON Z-AXIS
C     ISTRES=0, AXISYMMETRIC PROBLEM
C            =1, PLANE STRAIN PROBLEM
C            =2, PLANE STRESS PROBLEM
C
      IF(NCRACK.EQ.0) GO TO 20
      A=(R(NPJ)-R(NPI))**2+(Z(NPJ)-Z(NPI))**2
      IF(A.GT.0.0) GO TO 21
      I=NPJ
      J=NPK
      K=NPL
      L=NPI
      NPI=I
      NPJ=J
      NPK=K
      NPL=L
   21 CONTINUE
      KASE=4
      IF(R(NPI).NE.0.0) RETURN
      IF(ISTRES.EQ.0) KASE=6
      RETURN
   20 CONTINUE
      NI=NPI
      NJ=NPJ
      NK=NPK
      NL=NPL
      IF(R(NI).LT.R(NJ)) GO TO 1
      IF(R(NI).NE.R(NJ)) GO TO 2
      IF(Z(NI).LT.Z(NJ)) GO TO 1
    2 NI=NPJ
      NJ=NPI
      NPI=NI
      NPJ=NJ
    1 IF(R(NI).LT.R(NK)) GO TO 3
      IF(R(NI).NE.R(NK)) GO TO 4
      IF(Z(NI).LT.Z(NK)) GO TO 3
    4 NI=NPK
      NK=NPI
      NPI=NI
      NPK=NK
    3 IF(NPL.EQ.0) GO TU 5
      IF(R(NI).LT.R(NL)) GO TO 5
      IF(R(NI).NE.R(NL)) GO TO 6
      IF(Z(NI).LT.Z(NL)) GO TO 5
    6 NI=NPL
      NL=NPI
      NPI=NI
```

```
      NPL=NL
C
C     NODE POINT I IN PROPER LOCATION (CLOSEST TO ORIGIN)
C
    5 AJ=R(NJ)-R(NI)
      BJ=Z(NJ)-Z(NI)
      AK=R(NK)-R(NI)
      BK=Z(NK)-Z(NI)
      HJK=AJ*BK-AK*BJ
      IF(NPL.NE.0) GO TO 7
C
C     TRIANGULAR ELEMENT
C
      IF(HJK.GT.0.) GO TO 8
      NPK=NJ
      NPJ=NK
    8 IF(R(NPI).EQ.0.) GO TO 9
   10 KASE=1
      RETURN
    9 IF(ISTRES.NE.0) GO TO 10
      IF(R(NPK).EQ.0.) GO TO 11
      KASE=2
      RETURN
   11 KASE=3
      RETURN
C
C     RECTANGULAR ELEMENT
C
    7 AL=R(NL)-R(NI)
      BL=Z(NL)-Z(NI)
      HJL=AJ*BL-AL*BJ
      HKL=AK*BL-AL*BK
      IF(HJK.GT.0.) GO TO 12
      IF(HJL.GT.0.) GO TO 13
      IF(HKL.GT.0.) GO TO 14
      NPJ=NL
      NPL=NJ
      GO TO 15
   14 NPJ=NK
      NPK=NL
      NPL=NJ
      GO TO 15
   13 NPJ=NK
      NPK=NJ
      GO TO 15
   12 IF(HJL.GT.0.) GO TO 16
      NPJ=NL
      NPK=NJ
      NPL=NK
      GO TO 15
   16 IF(HKL.GT.0.) GO TO 15
      NPK=NL
      NPL=NK
   15 IF(R(NPI).EQ.0.) GO TO 17
   18 KASE=4
      RETURN
   17 IF(ISTRES.NE.0) GO TO 18
      IF(R(NPL).EQ.0.) GO TO 19
      KASE=5
      RETURN
```

```
   19 KASE=6
      RETURN
      END
C
C
C
      OVERLAY(MOHAN,2,0       )
      PROGRAM LNK1B
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1       NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2       KTAPE,KRJN,IPRINT,NUMST,MXSTRT,IELAST(20),IPLAST(20),
     3 WGT(20),NSTART(79),EI(5,20),IPELTP,INT,NPRCDS,IMPBX
C
      DIMENSION NPADJ(1600,8),NADJNP(1600),NADJEL(1600)
      MOHAN=5HMOHAN
      DO 5 I=1,NUMNP
      NADJNP(I)=0
      NADJEL(I)=0
      DO 5 J=1,MXADJP
    5 NPADJ(I,J)=0
C
      REWIND 1
      DO 7 M=1,NUMEL
      READ(1) NUME,IZONE,KASE,NPI,NPJ,NPK,NPL,NCRACK
    7 CALL ADJNP(MXADJP,MAXNP,NUMNP,NPADJ,NADJEL,NUME,NPI,NPJ,NPK,NPL)
      CALL VADJNP(MXADJP,NADJNP,MAXNP,NUMNP,NPADJ)
      REWIND 1
C
      REWIND 8
      WRITE(8) (NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP),I=1,NUMNP)
      REWIND 8
C
      IF((IPRINT.NE.1).AND.(IPRINT.NE.99))RETURN
C
      WRITE(6,1)
    1 FORMAT(1H1,38HTABLE OF ORIGINAL ADJACENT NODE POINTS//
     1 4X,4HNODE,13X,6HNO. OF,4X,6HNO. OF,27X,20HADJACENT NODE POINTS/
     2 4X,5HPOINT,11X,9HADJ. PTS.,1X,9HADJ. ELS.,5X,1H1,9X,1H2,9X,1H3,
     3 9X,1H4,9X,1H5,9X,1H9,9X,1H7,9X,1H8//)
      DO 2 I=1,NUMNP
    2 WRITE(6,3) I,NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP)
    3 FORMAT(I8,8X,2I10,8I10)
C
      RETURN
      END
C
C
C
      SUBROUTINE ADJNP(MXADJP,MAXNP,NUMNP,NPADJ,NADJEL,NUME,
     1NPI,NPJ,NPK,NPL)
      DIMENSION NPADJ(MAXNP,MXADJP),NADJEL(MAXNP),NA(4)
C
C**** FORM TABLE OF ADJACENT NODAL POINTS
C
C
C     MXADJP=MAX. NO. OF ADJACENT NODAL POINTS ALLOWED
C     MAXNP =MAX. NO. OF NODE POINTS
C     NUMNP =NO. OF NODE POINTS
C     NPADJ =ADJACENT NODE POINT NUMBER
C     NADJEL=NUMBER OF ADJACENT ELEMENTS AT EACH NODE POINT
```

```
C      NPI    =ELEMENT NODE POINT I
C      NPJ    =ELEMENT NODE POINT J
C      NPK    =ELEMENT NODE POINT K
C      NPL    =ELEMENT NODE POINT L, IF 0, TRIANGULAR ELEMENT
C      NUME   =ELEMENT NUMBER BEING CONSIDERED
C      NOTE- TABLE ASSUMED TO BE ALREADY ZEROED OUT
C
       NA(1)=NPI
       NA(2)=NPJ
       NA(3)=NPK
       NA(4)=NPL
       ICOUNT=1
     9 NPNUM=NA(1)
       NADJEL(NPNUM)=NADJEL(NPNUM)+1
       MX=NA(2)
       JCOUNT=1
     5 DO 1 I=1,MXADJP
       J=I
       IF(NPADJ(NPNUM,I).EQ.MX) GO TO 2
       IF(NPADJ(NPNUM,I).EQ.0)  GO TO 3
     1 CONTINUE
       WRITE (6,101)  NUME,NPNUM,MX,(NPADJ(NPNUM,I),I=1,MXADJP)
       CALL EXIT
C
     3 NPADJ(NPNUM,J)=MX
     2 JCOUNT=JCOUNT+1
       IF(JCOUNT.GT.3) GO TO 4
       IF(JCOUNT.GT.2) GO TO 102
       MX=NA(3)
       GO TO 5
   102 MX=NA(4)
       IF(MX.EQ.0) GO TO 4
       GO TO 5
C
     4 GO TO (6,7,8,103), ICOUNT
     6 ICOUNT=2
       NA(1)=NPJ
       NA(2)=NPK
       NA(3)=NPI
       NA(4)=NPL
       GO TO 9
C
     7 ICOUNT=3
       NA(1)=NPK
       NA(2)=NPI
       NA(3)=NPJ
       NA(4)=NPL
       GO TO 9
C
     8 ICOUNT=4
       NA(1)=NPL
       IF(NA(1).EQ.0) GO TO 103
       NA(2)=NPI
       NA(3)=NPJ
       NA(4)=NPK
       GO TO 9
C
   101 FORMAT(1H1,43HERROR IN FORMING ADJACENT NODAL POINT ARRAY/
      121H ELEMENT NUMBER     =,I5/21H NODE POINT NUMBER  =,I5/
      221H ADJACENT NODE POINT=,I5//15H NPADJ(NPNUM,I)/(21X,I5))
```

```
C
  103 RETURN
C
      END
C
C
C
      SUBROUTINE VADJNP(MXADJP,NADJNP,MAXNP,NUMNP,NPADJ)
      DIMENSION NADJNP(MAXNP),NPADJ(MAXNP,MXADJP)
C
C**** FORM V+CTOR INDICATING THE NUMBER OF ADJACENT NODE POINTS
C     AT EACH NODE POINT
C
C     MXADJP=MAX. NO. OF ADJACENT NODE POINTS ALLOWED
C     NADJNP8NO. OF ADJACENT NODE POINTS AT EACH NODE POINT
C     MAXNP =MAX. NO. OF NODE POINTS
C
C     NUMNP =NO. OF NODE POINTS
C     NPADJ =ADJACENT NODE POINT NUMBER
C
      DO 12 M=1,NUMNP
      DO 10 I=1,MXADJP
      J=I
      IF (NPADJ(M,I).EQ.0) GO TO 11
   10 CONTINUE
      NADJNP(M)=MXADJP
      GO TO 12
   11 NADJNP(M)=J-1
   12 CONTINUE
      RETURN
C
      END
C
C
C
      OVERLAY(MOHAN,3,0        )
      PROGRAM LNK3C
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1      NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2      KTAPE,KRJN,IPRINT,NUMST,MXSTRT,IELAST(20),IPLAST(20),
     3 WGT(20),NSTART(79),EI(5,20),IPELTP,INT,NPRCDS,IMPBX
C
      DIMENSION NPADJ(1600,8),NADJNP(1600),NADJEL(1600),NPTN(1600),
     1NPTP(1600),IGP(80),S(1600),NPLOW(80),NPHIGH(80),NPOUT(80),
     2NUMCP(80)
C
C
      EQUIVALENCE (NADJEL(1),NPLOW),(NADJEL(81),NPHIGH),(NADJEL(161),
     1NPOUT),(NADJEL(241),NUMCP)
C
      MOHAN=5HMOHAN
C
      REWIND 8
      READ(8) (NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP),I=1,NUMNP)
      REWIND 8
C
      MAXBO=0
      DO 5 I=1,NUMNP
      NUM=NADJNP(I)
      DO 5 J=1,NUM
```

```
      NP=NPADJ(I,J)
      NBAN=IABS(NP-I)
      IF(NBAN.LE.MAXBD) GO TO 6
      MAXBD=NBAN
    6 CONTINUE
C
      CALL PATH(MAXNP,NUMNP,NUMST,NSTART,NPTN,NPTP,MXADJP,NADJNP,NPADJ,
     1IGP,NJMGP)
C
      WRITE(6,8) NUMGP
    8 FORMAT( 1H1,23H NO. OF BASIC SEGMENTS=,I5//
     123H PARTITICN NEW NODE NO.//)
      WRITE(6,9) (I,IGP(I), I=1,NUMGP)
    9 FORMAT(2X,I5,10X,I5)
C
      WRITE(6,10) MAXBD
   10 FCRMAT(1H1/26H ORIGINAL HALF BAND WIDTH=,I5)
C
      MAXBD=0
      DO 11 I=1,NUMNP
      NPNEW=NPTP(I)
      NUM=NADJNP(I)
      DO 11 J=1,NUM
      MP=NPADJ(I,J)
      MPNEW=NPTP(MP)
      NBAN=IABS(NPNEW-MPNEW)
      IF(NBAN.LE.MAXBD) GO TO 11
      MAXBD=NBAN
   11 CCNTINJE
C
      WRITE(6,12) MAXBD
   12 FCRMAT(26H NEW       HALF BAND WIDTH=,I5)
C
      CALL MINI(MAXNP,NUMNP,NADJNP,MXADJP,NPADJ,NPTN,NPTP,S,MAXBD)
C
      IF(MAXBC.LT.MXNPB) GO TO 14
      WRITE(6,13)
   13 FCRMAT(//20H BANDWICTH TOO LARGE)
      CALL EXIT
C
   14 DO 15 I=1,NUMNP
      KN=NADJNP(I)
      DC 15 J=1,KN
      KT=NPADJ(I,J)
   15 NPADJ(I,J)=NPTP(KT)
C
      DO 1 I=1,NUMNP
      KP=NPTN(I)
    1 WRITE(8) I,NADJNP(KP),NADJEL(KP),(NPADJ(KP,J),J=1,MXADJP)
      REWIND 8
      DO 2 I=1,NUMNP
    2 READ(8) I,NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP)
C
      IF((IPRINT.NE.1).AND.(IPRINT.NE.99)) GO TO 19
      WRITE(6,16)
   16 FCRMAT(1H1,38HTABLE OF NEW       ADJACENT NODE POINTS//
     14X,4HNEW ,3X,4HOLD ,6X,6HNO. OF,4X,6HNO. OF,27X,
     220HADJ+CENT NODE POINTS/4X,4HNODE,3X,4HNODE,5X,9HADJ. PTS.,
     31X,9HADJ. ELS.,5X,1H1,9X,1H2,9X,1H3,
     4 9X,1H4,9X,1H5,9X,1H9,9X,1H7,9X,1H8//)
```

```
      DO 17 I=1,NUMNP
      KP=NPTN(I)
   17 WRITE(6,18) I,KP,NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP)
   18 FORMAT(2I8,2I10,8I10)
      GO TO 26
C
   19 IF(IPRINT.EQ.0) GO TO 26
      WRITE(6,20)
   20 FORMAT(1H1,21H NEW NUMBERING SCHEME//)
      DO 21 I=1,NUMNP,10
      IF((NUMNP-I).LT.10) IDUM=NUMNP
      IF((NUMNP-I).GE.10) IDUM=I+9
   23 WRITE(6,24) (J,J=I,IDUM)
   24 FORMAT(/20H ORIGINAL NODES     =,10I8)
   21 WRITE(6,25) (NPTP(J),J=I,IDUM)
   25 FORMAT( 20H NEW      NODES     =,10I8)
   26 CONTINUE
C
      CALL SIZE(MXCLS,NUMCLS,NPLOW,NPHIGH,NPOUT,NUMCP,NUMNP,MXADJP,
     1MXNPB,NADJNP,NPADJ,MAXNP)
C
      WRITE(8) NUMCLS,(NPLOW(I),NPHIGH(I),NPOUT(I),NUMCP(I),
     1I=1,NUMCLS)
      WRITE(8) (NPTN(I),NPTP(I),I=1,NUMNP)
      REWIND 8
C
      RETURN
      END
C
C
C
      SUBROUTINE PATH(MAXNP,NUMNP,NUMST,NSTART,NPTN,NPTP,MXADJP,
     1NADJNP,NPADJ,IGP,NUMGP)
C
      DIMENSION NSTART(NUMST),NPTN(MAXNP),NPTP(MAXNP),NADJNP(MAXNP),
     1NPADJ(MAXNP,MXADJP),IGP(1)
C
C      NUMST =NO. OF START NODES
C      NSTART=STARTING NODE NUMBERS
C      NPTN  =OLD NODE NOS. IN NEW ORDER
C      NPTP  =NEW NODE NOS. IN OLD ORDER
C      IGP   =LAST NODE IN PARTITION
C      NUMGP =NO. OF PARTITIONS
C
      KOUNT=0
      IN=1
      DO 1 I=1,NUMNP
      NPTN(I)=0
    1 NPTP(I)=0
C
      DO 2 I=1,NUMST
      NP=NSTART(I)
      NPTP(NP)=IN
      KOUNT=KOUNT+1
    2 NPTN(KOUNT)=NP
C
      IGP(IN)=KOUNT
C
    4 DO 7 I=1,NUMNP
```

```
      IF(NPTP(I).NE.IN) GO TO 7
      NUM=NADJNP(I)
      DO 3 J=1,NUM
      NP=NPADJ(I,J)
      IF(NPTP(NP).NE.0) GO TO 3
      NPTP(NP)=IN+1
      KOUNT=KOUNT+1
      NPTN(KOUNT)=NP
      IF(KOUNT .EQ.NUMNP) GO TO 5
    3 CONTINUE
    7 CONTINUE
C
      IN=IN+1
      IGP(IN)=KOUNT
      GO TO 4
C
    5 IGP(IN+1)=KOUNT
      NUMGP=IN+1
C
      DO 6 I=1,NUMNP
      NPOLD=NPTN(I)
    6 NPTP(NPOLD)=I
C
      RETURN
      END
C
C
C
      SUBROUTINE MINI(MAXNP,NUMNP,NADJNP,MXADJP,NPADJ,NPTN,NPTP,S,
     1MAXBDP)
C
      DIMENSION NADJNP(MAXNP),NPADJ(MAXNP,MXADJP),NPTN(MAXNP),
     1NPTP(MAXNP),S(MAXNP)
C
C     S      =VECTOR OF WEIGHTING FACTORS
C     MAXBDP=MAX. PREVIOUS BANDWIDTH
C     MAXBD =MAX. BANDWIDTH
C
C     COMPUTE WEIGHTING FACTORS FOR OLD ORDER
C
    3 DO 1 I=1,NUMNP
      S(I)=FLOAT(I)
      NPOLD=NPTN(I)
      NUM=NADJNP(NPOLD)
      DO 2 J=1,NUM
      NADJ=NPADJ(NPOLD,J)
      NPNEW=NPTP(NADJ)
    2 S(I)=S(I)+FLOAT(NPNEW)
    1 S(I)=S(I)/FLOAT(NUM+1)
C
C     SORT S VECTOR AND REORDER NODES
C
      CALL SORT1(S,NPTN,MAXNP,NUMNP,1,1,1,1)
C
C     C-MPUTE BANDWIDTH OF NEW ORDER
C
      MAXBD=0
      DO 14 I=1,NUMNP
      NPOLD=NPTN(I)
      NUM=NADJNP(NPOLD)
```

```
      DO 10 J=1,NUM
      NADJ=NPADJ(NPOLD,J)
      DO 11 K=1,NUMNP
      KK=K
      IF(NADJ.EQ.NPTN(K)) GO TO 15
   11 CONTINUE
      WRITE(6,100) I,NPOLD,(NPTN(L),L=1,NUMNP)
  100 FORMAT(1H1,13HERROR IN MINI//10X,2I10//(10X,10I10))
      CALL EXIT
   15 NPNEW=KK
      NBAN=IABS(NPNEW-I)
      IF(MAXBD.LT.NBAN) MAXBD=NBAN
   10 CONTINUE
   14 CONTINUE
C
      WRITE(6,6) MAXBD
    6 FORMAT(26H NEW       HALF BAND WIDTH=,I5)
C
      IF(MAXBDP.LE.MAXBD) GO TO 12
C
      DO 5 I=1,NUMNP
      NPOLD=NPTN(I)
    5 NPTP(NPOLD)=I
      MAXBDP=MAXBD
      GO TO 3
C
   12 DO 16 I=1,NUMNP
      NPNEW=NPTP(I)
   16 NPTN(NPNEW)=I
      RETURN
      END
C
C
C
      SUBROUTINE SORT1(IARRAY,JARRAY,MXRCDS,NRECDS,
     1IWRDS,JWRDS,IKEY,ISWT)
C
      DIMENSION IARRAY(MXRCDS,IWRDS ),JARRAY(MXRCDS,JWRDS)
C
C     IARRAY=ARRAY TO BE SORTED
C     JARRAY=ASSOCIATED [illegible]AT MAY BE SORTED AS IARRAY
C     MXRCDS=MAX. NO. [illegible] IN ARRAYS
C     NRECDS=NO. OF RECO[illegible] SORTED
C     IWRDS =WORDS PER REC[illegible] IARRAY
C     JWRDS =WORDS PER RECORD FOR JARRAY
C     IKEY  =LOCATION IN IARRAY RECORD OF SORT WORD
C     ISWT  =0 ONLY SORT IARRAY
C           =1 ALSO SORT JARRAY
C
      LOGICAL CHECK
      M=NRECDS-1
C
    1 CHECK=.FALSE.
C
      DO 6 I=1,2
C
      DO 2 J=I,M,2
C
      IF(IARRAY(J,IKEY).LE.IARRAY(J+1,IKEY)) GO TO 2
C
```

```
      DO 3  K=1,IWRDS
      ITEMP=IARRAY(J,K)
      IARRAY(J,K)=IARRAY(J+1,K)
    3 IARRAY(J+1,K)=ITEMP
C
      IF(ISWT.EQ.0) GO TO 5
      DO 4  K=1,JWRDS
      JTEMP=JARRAY(J,K)
      JARRAY(J,K)=JARRAY(J+1,K)
    4 JARRAY(J+1,K)=JTEMP
C
    5 CHECK=.TRUE.
C
    2 CONTINUE
    6 CONTINUE
C
      IF(CHECK) GO TO 1
      RETURN
      END
C
C
C
      SUBROUTINE SIZE(MXCLS,NUMCLS,NPLOW,NPHIGH,NPOUT,NUMCP,NUMNP,
     1MXADJP,MXNPB,NADJNP,NPADJ,MAXNP)
C
      DIMENSION NPLOW(MXCLS),NPHIGH(MXCLS),NPOUT(MXCLS),NUMCP(MXCLS)
C
      DIMENSION NADJNP(MAXNP),NPADJ(MAXNP,MXADJP)
C
      DO 1  I=1,MXCLS
      NPLOW(I)=0
      NPHIGH(I)=0
      NPOUT(I)=0
    1 NUMCP(I)=0
C
      ICOUNT=1
      NPOUT(1)=0
      NPLOW(1)=1
C
      NP=1
C
    8 MPLRG=0
      MPSML=NUMNP
      NPLRG=0
      NPSML=NUMNP
C
    7 NUM=NADJNP(NP)
      LPLRG=0
      LPSML=NUMNP
      DO 2  J=1,NUM
      NPNUM=NPADJ(NP,J)
      IF(NPNUM.GT.NPLRG) NPLRG=NPNUM
      IF(NPNUM.GT.LPLRG) LPLRG=NPNUM
      IF(NPNUM.LT.NPSML) NPSML=NPNUM
      IF(NPNUM.LT.LPSML) LPSML=NPNUM
    2 CONTINUE
C
      IF((LPLRG-LPSML+1).LE.MXNPB) GO TO 3
C
      IDUMMY=LPLRG-LPSML+1
```

```
      WRITE(6,103) NP,IDUMMY
  103 FORMAT(1H1,13HERROR IN SIZE/29H BAND WIDTH TOO LARGE AT NODE,
     1I5/12H BAND WIDTH=,I5)
      CALL EXIT
C
    3 IF((NPLRG-NPSML+1).GT.MXNPB) GO TO 4
C
      IF(MPLRG.LT.NPLRG) MPLRG=NPLRG
      IF(MPSML.GT.NPSML) MPSML=NPSML
      IF(NP.GE.NUMNP) GO TO 5
      NP=NP+1
      GO TO 7
C
    5 NPOUT(ICOUNT)=MPSML-1
      NUMCP(ICOUNT)=MPLRG
      NPHIGH(ICOUNT)=NP
    6 NUMCLS=ICOUNT
C
      DO 9 I=1,NUMCLS
      IF(NPOUT(I).GE.NPLOW(I)) NPOUT(I)=NPLOW(I)-1
      IF(NUMCP(I).LT.NPHIGH(I)) NUMCP(I)=NPHIGH(I)
    9 CONTINUE
C
      RETURN
C
    4 NPOUT(ICOUNT)=MPSML-1
      NUMCP(ICOUNT)=MPLRG
      NPHIGH(ICOUNT)=NP-1
      ICOUNT=ICOUNT+1
      IF(ICOUNT.LE.MXCLS) GO TO 101
      WRITE(6,102) NP,NPHIGH(ICOUNT),NUMNP,ICOUNT
  102 FORMAT(1H1,25HTOO MANY CLUSTERS IN SIZE//
     110X,4I10)
      CALL EXIT
C
  101 NPLOW(ICOUNT)=NPHIGH(ICOUNT-1)+1
      GO TO 8
C
C
C
C
      END
C
C
C
      OVERLAY(MOHAN,4,0      )
      PROGRAM LNK1D
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1       NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2       KTAPE,KRJN,IPRINT,NUMST,MXSTRT,IELAST(20),IPLAST(20),
     3 WGT(20),NSTART(79),EI(5,20),IPELTP,INT,NPRCDS,IMPBX
C
      DIMENSION NPTP(1600),NPTN(1600)
      DIMENSION NPLOW(80),NPHIGH(80),NPOUT(80),NUMCP(80),NELCLS(80)
     1,NMPCLS(80)
      DIMENSION TMP(15000),NTMP(15000),SS(10),HS(10)
C
      EQUIVALENCE (TMP,NTMP)
      MOHAN=5HMOHAN
      REWIND 1
```

```
      REWIND 3
      REWIND 8
      DO 1 I=1,NUMNP
    1 READ(8) I,NTMP(I),NTMP(I),(NTMP(J),J=1,MXADJP)
      READ(8) NUMCLS,(NPLOW(I),NPHIGH(I),NPOUT(I),NUMCP(I),
     1I=1,NUMCLS)
      READ(8) (NPTN(I),NPTP(I),I=1,NUMNP)
      REWIND 8
C
      DO 27 NN=1,NUMEL
      READ(1) N,IZ,KASE,I,J,K,L,NC
      I=NPTP(I)
      J=NPTP(J)
      K=NPTP(K)
      IF(L.EQ.0) GO TO 29
      L=NPTP(L)
      KEY=MIN0(I,J,K,L)
      GO TO 27
   29 KEY=MIN0(I,J,K)
   27 WRITE(3) KEY,N,IZ,KASE,I,J,K,L,NC
      MXRCDS=15000/9
      NWRDS=9
      CALL GSORT(NTMP,NUMEL,NWRDS,1,MXRCDS,3,1,4,12)
C
      REWIND 14
      REWIND 3
      DO 30 N=1,NUMNP
      READ(14) I,R,D,IT,TH
      NP=NPTP(I)
   30 WRITE(3) NP,R,D,IT,TH
      MXRCDS= 15000/5
      NWRDS=5
      CALL GSORT(NTMP,NUMNP,NWRDS,1,MXRCDS,3,4,10,12)
C
      DO 31 I=1,NUMNP
   31 READ(4) N,R,D,IT,TH
      READ(14) NZONES
      WRITE(4) NZONES
      DO 32 NN=1,NZONES
      READ(14)   I,IE,IP,W,(EI(J),J=1,5)
      WRITE(4)   I,IE,IP,W,(EI(J),J=1,5)
      IF(IP.EQ.0) GO TO 32
      IF(IP.GT.1) GO TO 33
      READ(14)    N,(SS(J),J=1,N),(HS(J),J=1,N)
      WRITE(4)    N,(SS(J),J=1,N),(HS(J),J=1,N)
      GO TO 32
   33 IF(IP.GT.2) GO TO 48
      READ(14)   A,B,C
      WRITE(4)   A,B,C
      GO TO 32
   48 IF(IP.GT.3) GO TO 34
      READ(14)A,B,C,D,E,F,MYIELD,IRESID,JTENSN
      WRITE(4)A,B,C,D,E,F,MYIELD,IRESID,JTENSN
      GO TO 32
   34 WRITE(6,35) IP,N
   35 FORMAT(1H1,21HERROR   ) L1C, IPLAST=,I5,9H FOR ZONE,I5)
      CALL EXIT
   32 CONTINUE
C
      DO 40 I=1,MXCLS
```

```
40 NELCLS(I)=0
   NUMELP=0
   IF(NUMPEL.EQ.0) GO TO 39
   REWIND 3
   DO 36 NN=1,NUMPEL
   READ(14) N,IZ,IP,I,J,K,L,NC,ITI,ITJ,ITK,ITL,THI,THJ,THK,THL,
  1RI,RJ,RK,RL,ZI,ZJ,ZK,ZL
   I=NPTP(I)
   J=NPTP(J)
   K=NPTP(K)
   IF(L.EQ.0) GO TO 37
   L=NPTP(L)
37 DO 41 JJ=1,NUMCLS
   IF((I.GE.NPLOW(JJ)).AND.(I.LE.NPHIGH(JJ))) GO TO 42
   IF((J.GE.NPLOW(JJ)).AND.(J.LE.NPHIGH(JJ))) GO TO 42
   IF((K.GE.NPLOW(JJ)).AND.(K.LE.NPHIGH(JJ))) GO TO 42
   IF((L.GE.NPLOW(JJ)).AND.(L.LE.NPHIGH(JJ))) GO TO 42
   GO TO 41
42 WRITE(3) JJ,N,IZ,IP,I,J,K,L,NC,ITI,ITJ,ITK,ITL,THI,THJ,THK,
  1THL,RI,RJ,RK,RL,ZI,ZJ,ZK,ZL
   NUMELP=NUMELP+1
   NELCLS(JJ)=NELCLS(JJ)+1
41 CONTINUE
36 CONTINUE
   REWIND 3
   REWIND 14
   MXRCDS=15000/25
   NWRDS=25
   CALL GSORT(NTMP,NUMELP,NWRDS,1,MXRCDS,3,14,10,12)
   REWIND 14
   DO 38 NN=1,NUMELP
   READ(14) JJ,N,IZ,IP,I,J,K,L,NC,ITI,ITJ,ITK,ITL,
  1THI,THJ,THK,THL,RI,RJ,RK,RL,ZI,ZJ,ZK,ZL
38 WRITE(4) JJ,N,IZ,IP,I,J,K,L,NC,ITI,ITJ,ITK,ITL,
  1THI,THJ,THK,THL,RI,RJ,RK,RL,ZI,ZJ,ZK,ZL
39 REWIND 14
   REWIND 4
   DO 47 IC=1,NUMCLS
   KC=0
   IF((NELCLS(IC)/MXPELB)*MXPELB.LT.NELCLS(IC)) KC=1
   IF(NELCLS(IC).EQ.0) KC=0
   NMPCLS(IC)=NELCLS(IC)/MXPELB+KC
47 CONTINUE
   DO 43 I=1,NUMNP
43 READ(8) I,NTMP(I),NTMP(I),(NTMP(J),J=1,MXADJP)
   WRITE(8) NUMCLS,(NPLOW(I),NPHIGH(I),NPOUT(I),NUMCP(I),NELCLS(I),
  1NMPCLS(I),I=1,NUMCLS)
   WRITE(8) (NPTN(I),NPTP(I),I=1,NUMNP)
   REWIND 8
   WRITE(6,44) NUMCLS
44 FORMAT(1H1,10HCLUSTERING//10X,16HNO. OF CLUSTERS=,I5//
  110X,5HNPLOW,5X,6HNPHIGH,4X,5HNPOUT,5X,5HNUMCP,5X,6HNELCLS,5X,
  26HNMPCLS//)
   WRITE(6,45) (NPLOW(I),NPHIGH(I),NPOUT(I),NUMCP(I),NELCLS(I),
  1NMPCLS(I),I=1,NUMCLS)
45 FORMAT(5X,6I10)
C
   WRITE(6,46) NUMELP
46 FORMAT(//35H NO. OF NONLINEAR ELEMENTS ON TAPE =,I5)
C
```

```
      RETURN
      END
C
C
      SUBROUTINE GSORT(IARRAY,NRCDS,NWRDS,NKEY,MXRCDS,
     1 INTAPE,IOUTAP,INT1,INT2)
C
      DIMENSION IARRAY(MXRCDS,NWRDS)
C
C     IARRAY=BUFFER STORAGE REGION
C     NRCDS =NO. OF RECORDS IN ARRAY TO BE SORTED
C     NWRDS =NO. OF WORDS PER RECORD
C     NKEY  =LOCATION OF WORD TO BE SORTED ON
C     MXRCDS=MAX. SIZE OF BUFFER REGION AVAILABLE (INPUT)
C     INTAPE=INPUT TAPE WITH ORIGINAL DATA
C     IOUTAP=OUTPUT TAPE WITH REORDERED DATA
C     INT   =INTERMEDIATE TAPES
C
      REWIND INTAPE
      REWIND IOUTAP
      REWIND INT1
      REWIND INT2
C
      IF(NRCDS.GT.MXRCDS) GO TO 1
C
C**** INTERNAL SORT ONLY REQUIRED
C
      DO 2 I=1,NRCDS
    2 READ(INTAPE) (IARRAY(I,J),J=1,NWRDS)
      CALL SORT2(IARRAY,IARRAY,MXRCDS,NRCDS,NWRDS,NWRDS,NKEY,0)
      DO 3 I=1,NRCDS
    3 WRITE(IOUTAP) (IARRAY(I,J),J=1,NWRDS)
      REWIND INTAPE
      REWIND IOUTAP
      RETURN
C
C**** TAPE SORT ROUTINE REQUIRED
C
    1 IXRCDS=(MXRCDS/4)
      CALL TSORT(IARRAY,NRCDS,IXRCDS,NWRDS,NKEY,
     1 INTAPE,IOUTAP,INT1,INT2)
      RETURN
      END
C
C
C
      SUBROUTINE SORT2(IARRAY,JARRAY,MXRCDS,NRECDS,
     1IWRDS,JWRDS,IKEY,ISWT)
C
      DIMENSION IARRAY(MXRCDS,IWRDS ),JARRAY(MXRCDS,JWRDS)
C
C     IARRAY=ARRAY TO BE SORTED
C     JARRAY=ASSOCIATED ARRAY THAT MAY BE SORTED AS IARRAY
C     MXRCDS=MAX. NO. OF RECORDS IN ARRAYS
C     NRECDS=NO. OF RECORDS TO BE SORTED
C     IWRDS =WORDS PER RECORD FOR IARRAY
C     JWRDS =WORDS PER RECORD FOR JARRAY
C     IKEY  =LOCATION IN IARRAY RECORD OF SORT WORD
C     ISWT  =0 ONLY SORT IARRAY
C           =1 ALSO SORT JARRAY
```

```
C
      LOGICAL CHECK
      M=NRECDS-1
C
    1 CHECK=.FALSE.
C
      DO 6 I=1,2
C
      DO 2 J=I,M,2
C
      IF(IARRAY(J,IKEY).LE.IARRAY(J+1,IKEY)) GO TO 2
C
      DO 3 K=1,IWRDS
      ITEMP=IARRAY(J,K)
      IARRAY(J,K)=IARRAY(J+1,K)
    3 IARRAY(J+1,K)=ITEMP
C
      IF(ISWT.EQ.0) GO TO 5
      DO 4 K=1,JWRDS
      JTEMP=JARRAY(J,K)
      JARRAY(J,K)=JARRAY(J+1,K)
    4 JARRAY(J+1,K)=JTEMP
C
    5 CHECK=.TRUE.
C
    2 CONTINUE
    6 CONTINUE
C
      IF(CHECK) GO TO 1
      RETURN
      END
C
C
C
      SUBROUTINE TSORT(IARRAY,NRCDS,IXRCDS,NWRDS,NKEY,
     1INTAPE,IOUTAP,INT1,INT2)
C
      DIMENSION IARRAY(IXRCDS,NWRDS,4),CHECK(2),ISWT(2),IOUT(2),JNUM(2)
C
      LOGICAL CHECK
C
C**** READ INTAPE,SORT GROUPS,AND SPLIT ONTO INT1 AND INT2
C
      IRCDS=0
      ISWC=0
      JNUM(1)=0
      JNUM(2)=0
      KTAPE=INT1
    3 IF((IRCDS+IXRCDS).LE.NRCDS) KN=IXRCDS
      IF((IRCDS+IXRCDS).GT.NRCDS) KN=NRCDS-IRCDS
      DO 1 I=1,KN
    1 READ(INTAPE) (IARRAY(I,J,1),J=1,NWRDS)
      IRCDS=IRCDS+KN
      CALL SORT2(IARRAY,IARRAY,IXRCDS,KN,NWRDS,NWRDS,NKEY,0)
      WRITE(KTAPE) KN,((IARRAY(I,J,1),J=1,NWRDS),I=1,KN)
      IF(ISWC.EQ.0) GO TO 2
      ISWC=0
      JNUM(2)=JNUM(2)+1
      KTAPE=INT1
      GO TO 4
```

```
    2 ISWC=1
      JNUM(1)=JNUM(1)+1
      KTAPE=INT2
    4 IF(IRCDS.LT.NRCDS) GO TO 3
C
C**** SORT RECORD CLUSTERS ON INT1 AND INT2
C
      REWIND INTAPE
      REWIND IOUTAP
      REWIND INT1
      REWIND INT2
      DO 5 I=1,2
    5 ISWT(I)=0
      KP1=INT1
      KP2=INTAPE
      LP1=INT2
      LP2=IOUTAP
      ICOUNT=2
C
C
  900 I=1
C
      CHECK(1)=.FALSE.
      CHECK(2)=.FALSE.
C
  902 IF(ICOUNT.EQ.1) GO TO 800
      ICOUNT=1
      GO TO 901
  800 ICOUNT=2
C
  901 DO 100 NC=1,2
C
      IF(ICOUNT.EQ.2) GO TO 6
C
      IF(NC.EQ.2) GO TO 7
      KT1=KP1
      KT2=KP2
      GO TO 8
C
    7 KT1=LP1
      KT2=LP2
      GO TO 8
C
    6 IF(NC.EQ.2) GO TO 9
      KT1=KP2
      KT2=KP1
      GO TO 8
C
    9 KT1=LP2
      KT2=LP1
C
    8 IF(ISWT(NC).EQ.1) GO TO 100
C
C**** READ FIRST TWO SORTED CLUSTERS
C
      J=1
      READ(KT1) KN1,((IARRAY(K,L,1),L=1,NWRDS),K=1,KN1)
      IF(I.EQ.1) GO TO 200
      WRITE(KT2) KN1,((IARRAY(K,L,1),L=1,NWRDS),K=1,KN1)
      J=2
```

```
      READ(KT1) KN1,((IARRAY(K,L,1),L=1,NWRDS),K=1,KN1)
  200 J=J+1
      READ(KT1) KN2,((IARRAY(K,L,2),L=1,NWRDS),K=1,KN2)
C
C**** SORT TWO CLUSTERS
C
  201 K=1
      L=1
      M=1
      M2=KN1+KN2
C
  207 IF(IARRAY(K,NKEY,1).LE.IARRAY(L,NKEY,2)) GO TO 202
      CHECK(NC)=.TRUE.
      IF(M.GT.IXRCDS) GO TO 203
      M1=M
      IS=3
      GO TO 204
  203 M1=M-IXRCDS
      IS=4
  204 DO 205 N=1,NWRDS
  205 IARRAY(M1,N,IS)=IARRAY(L,N,2)
      M=M+1
      IF(M.GT.M2) GO TO 300
      L=L+1
      IF(L.LE.KN2) GO TO 207
      DO 208 KM=K,KN1
      IF(M.GT.IXRCDS) GO TO 209
      M1=M
      IS=3
      GO TO 210
  209 M1=M-IXRCDS
      IS=4
  210 DO 211 N=1,NWRDS
  211 IARRAY(M1,N,IS)=IARRAY(KM,N,1)
  208 M=M+1
      GO TO 300
C
  202 IF(M.GT.IXRCDS) GO TO 212
      M1=M
      IS=3
      GO TO 213
  212 M1=M-IXRCDS
      IS=4
  213 DO 214 N=1,NWRDS
  214 IARRAY(M1,N,IS)=IARRAY(K,N,1)
      M=M+1
      IF( M.GT.M2) GO TO 300
      K=K+1
      IF(K.LE.KN1) GO TO 207
      DO 215 LM=L,KN2
      IF(M.GT.IXRCDS) GO TO 216
      M1=M
      IS=3
      GO TO 217
  216 M1=M-IXRCDS
      IS=4
  217 DO 218 N=1,NWRDS
  218 IARRAY(M1,N,IS)=IARRAY(LM,N,2)
  215 M=M+1
      GO TO 300
```

```
C
C**** WRITE TWO MERGED ARRAYS ONTO 2ND TAPE
C
  300 WRITE(KT2) KN1,((IARRAY(K,L,3),L=1,NWRDS),K=1,KN1)
      WRITE(KT2) KN2,((IARRAY(K,L,4),L=1,NWRDS),K=1,KN2)
C
      IF(J.GE.(JNUM(NC)-1)) GO TO 219
      READ(KT1) KN1,((IARRAY(K,L,1),L=1,NWRDS),K=1,KN1)
      READ(KT1) KN2,((IARRAY(K,L,2),L=1,NWRDS),K=1,KN2)
      J=J+2
      GO TO 201
C
  219 IOUT(NC)=KT2
      IF(J.EQ.JNUM(NC)) GO TO 220
      READ(KT1)  KN1,((IARRAY(K,L,1),L=1,NWRDS),K=1,KN1)
      WRITE(KT2) KN1,((IARRAY(K,L,1),L=1,NWRDS),K=1,KN1)
  220 REWIND KT1
      REWIND KT2
C
  100 CONTINUE
C
      I=I+1
C
      IF(JNUM(1).EQ.2) ISWT(1)=1
      IF(JNUM(2).EQ.2) ISWT(2)=1
C
      IF(I.EQ.2) GO TO 902
C
      IF(.NOT.CHECK(1)) ISWT(1)=1
      IF(.NOT.CHECK(2)) ISWT(2)=1
      IF((ISWT(1).EQ.1).AND.(ISWT(2).EQ.1)) GO TO 101
      GO TO 900
C
C**** MUST NOW MERGE THE TWO ORDERED TAPES
C
  101 KP1=IOUT(1)
      KP2=IOUT(2)
      REWIND INTAPE
      REWIND IOUTAP
      REWIND INT1
      REWIND INT2
      IF(KP1.NE.INTAPE) GO TO 103
      IF(KP2.NE.IOUTAP) GO TO 102
  106 INTER=INT1
      GO TO 400
  102 INTER=IOUTAP
      GO TO 400
  103 IF(KP1.NE.IOUTAP) GO TO 105
      IF(KP2.NE.INTAPE) GO TO 104
      GO TO 106
  104 INTER=INTAPE
      GO TO 400
  105 IF(KP1.NE.INT1) GO TO 107
      IF(KP2.NE.INTAPE) GO TO 108
      GO TO 102
  108 IF(KP2.NE.IOUTAP) GO TO 102
      INTER=INTAPE
      GO TO 400
  107 IF(KP2.NE.IOUTAP) GO TO 102
      GO TO 106
```

```
C
C      FIRST TAPE IS KP1 , HAS JNUM(1) CLUSTERS
C      SECOND TAPE IS KP2, HAS JNUM(2) CLUSTERS
C      MERGED TAPE IS INTER
C
  400 J1=1
      J2=1
  405 READ(KP1) KN1,((IARRAY(I,J,1),J=1,NWRDS),I=1,KN1)
      READ(KP2) KN2,((IARRAY(I,J,2),J=1,NWRDS),I=1,KN2)
C
      K=1
      L=1
  402 IF(IARRAY(K,NKEY,1).LE.IARRAY(L,NKEY,2)) GO TO 401
      WRITE(INTER) (IARRAY(L,N,2),N=1,NWRDS)
      L=L+1
      IF(L.LE.KN2) GO TO 402
      DO 403 KN=K,KN1
  403 WRITE(INTER)  (IARRAY(KN,N,1),N=1,NWRDS)
      GO TO 500
  401 WRITE(INTER) (IARRAY(K,N,1),N=1,NWRDS)
      K=K+1
      IF(K.LE.KN1) GO TO 402
      DO 404 LN=L,KN2
  404 WRITE(INTER)  (IARRAY(LN,N,2),N=1,NWRDS)
      GO TO 500
C
C
  500 J1=J1+1
      J2=J2+1
      IF((J1.LE.JNUM(1)).AND.(J2.LE.JNUM(2))) GO TO 405
      IF((J1.GT.JNUM(1)).AND.(J2.GT.JNUM(2))) GO TO 600
C
      IF(J2.GT.JNUM(2)) GO TO 501
  503 READ(KP2) KN2,((IARRAY(I,J,2),J=1,NWRDS),I=1,KN2)
      DO 502 I=1,KN2
  502 WRITE(INTER) (IARRAY(I,N,2),N=1,NWRDS)
      J2=J2+1
      IF(J2.GT.JNUM(2)) GO TO 600
      GO TO 503
C
  501 READ(KP1) KN1,((IARRAY(I,J,1),J=1,NWRDS),I=1,KN1)
      DO 504 I=1,KN1
  504 WRITE(INTER) (IARRAY(I,N,1),N=1,NWRDS)
      J1=J1+1
      IF(J1.GT.JNUM(1)) GO TO 600
      GO TO 501
C
  600 REWIND KP1
      REWIND KP2
      REWIND INTER
      IF(INTER.EQ.IOUTAP) RETURN
      DO 601 I=1,NRCDS
      READ(INTER)(IARRAY(1,J,1),J=1,NWRDS)
  601 WRITE(IOUTAP) (IARRAY(1,J,1),J=1,NWRDS)
      REWIND INTER
      REWIND IOUTAP
      RETURN
C
C
C
```

```
      END
C
C
C
      OVERLAY(MOHAN,5,0      )
      PROGRAM LNK1E
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1        NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2        KTAPE,KRJN,IPRINT,NUMST,MXSTRT,IELAST(20),IPLAST(20),
     3 WGT(20),NSTART(79),EI(5,20),IPELTP,INT,NPRCDS,IMPBX
C
      DIMENSION NPTN(1600),NADJNP(350),NPADJ(350,8),NADJEL(350)
     1,R(350),Z(350),ITYPE(350),THETA(350),XMASS(350)
C
      DIMENSION C(4,4),CK(8,8),AINT(23)
C
C
      DIMENSION SNPUJ(350),SNPUW(350),SNPWW(350),SADLU(350,8)
     1,SADUW(350,8),SADWU(350,8),SADWW(350,8)
C
      MCHAN=5HMOHAN
C
      REWIND 1
      REWIND 4
      REWIND 8
      REWIND 10
      REWIND 12
C
C
      KEND=0
      NPOUT=0
      NUMCP=0
      NUMNPB=0
      NPR=MXNPB
      KX=1
C
C     KEND  =INDICATES END OF ELEMENT DATA
C     NPOUT =NUM. OF COMPLETE NODE POINTS OUT ON TAPE
C     NUMCP =NUM. OF COMPLETE NODE POINTS
C     NUMNPB=NUM. OF NODE POINTS IN BUFFER
C     NPR   =NO. OF NODES REMAINING TO BE COMPLETED IN BUFFER
C     KX    =BEGINNING OF REGION TO BE ZEROED OUT
C
C**** ZERO OUT BUFFER REGION
C
    1 ISWTCH=1
      GO TO 900
C
C**** READ IN FIRST RECORDS
C
    2 IF(NUMNP.LT.MXNPB) NUMNPB=NUMNP
      IF(NUMNP.GE.MXNPB) NUMNPB=MXNPB
      DO 3 I=1,NUMNPB
      READ(8) NPN,NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP)
      READ(4) NPN,R(I),Z(I),ITYPE(I),THETA(I)
    3 CONTINUE
C
      ICOUNT=0
    4 READ(1)  KEY,NUME,IZONE,KASE,NTI,NTJ,NTK,NTL,NCRACK
      ICOUNT=ICOUNT+1
```

```
      LNP=MAX0(NTI,NTJ,NTK,NTL)
C
      IF((LNP-NPOUT).GT.MXNPB) GO TO 100
C
C     SUFFICIENT ROOM IN BUFFER REGION
C
    6 NPI=NTI-NPOUT
      NPJ=NTJ-NPOUT
      NPK=NTK-NPOUT
      IF(NTL.EQ.0) NPL=0
      IF(NTL.NE.0) NPL=NTL-NPOUT
      S1=0.0
      C1=0.0
      IE=IELAST(IZONE)
      A1=EI(1,IZONE)
      A2=EI(2,IZONE)
      A3=EI(3,IZONE)
      A4=EI(4,IZONE)
      A5=EI(5,IZONE)
      RHO=WGT(IZONE)/(386.4*1728.)
      CALL ELAST(IE,ISTRES,A1,A2,A3,A4,A5,C,NUME)
      CALL STIFF(KASE,NPI,NPJ,NPK,NPL,NUME,MXNPB,ISTRES,C,R,Z,CK,AINT,
     1S1,C1,NCRACK)
      CALL ADJUSK(MXNPB,CK,ITYPE,THETA,NPI,NPJ,NPK,NPL)
      CALL DISTK(MXNPB,MXADJP,CK,SNPUU,SNPUW,SNPWW,SADUU,SADUW,SADWU,
     1SADWW,NPI,NPJ,NPK,NPL,NPADJ,NPOUT)
      IF(NCRACK.EQ.1) GO TO 5
      CALL MASS(MXNPB,RHO,R,Z,AINT,XMASS,S1,C1,NPI,NPJ,NPK,NPL,ISTRES)
    5 WRITE(12) KEY,NUME,IZONE,NTI,NTJ,NTK,NTL,NCRACK,
     1((C(I,J),I=1,4),J=1,4),KASE,S1,C1
C
      IF(ICOUNT.LT.NUMEL) GO TO 4
      KEND=1
      KEY=NUMNP+1
C
C**** INSUFFICIENT ROOM IN BUFFER REGION
C
  100 NUMCP=KEY-1
      NUMNPB=NUMCP-NPOUT
C
C     PRINT STIFFNESS TABLES
C
  113 CALL        PRNK(MXNPB,MXADJP,NADJNP,NPADJ,NADJEL,IPRINT,SNPUU,
     1SNPUW,SNPWW,SADUU,SADUW,SADWU,SADWW,THETA,ITYPE,XMASS,NPOUT,
     2NUMNPB)
C
C     WRITE STIFFNESS TABLES ONTO TAPE 10 WITH MASS VECTOR
C
      DO 101 I=1,NUMNPB
  101 WRITE(10) I    ,NADJNP(I),ITYPE(I),THETA(I),XMASS(I),SNPUU(I),
     1SNPUW(I),SNPWW(I),(NPADJ(I,J),SADUU(I,J),SADUW(I,J),SADWU(I,J),
     2 SADWW(I,J),J=1,MXADJP)
C
      IF(KEND.EQ.1) RETURN
      NPR=MXNPB-NUMNPB
      GO TO 902
C
C     ZERO REMAINING BUFFER AREA
C
  107 KX=NPR+1
```

```
      ISWTCH=2
      GC TO 900
C
C     READ IN REMAINING NODE POINT AND ADJACENCY DATA TO FILL IN BUFFER
C
  108 IF((NUMNP-NJMCP).LT.MXNPB) KNP=NUMNP-NUMCP-NPR
      IF((NUMNP-NUMCP).GE.MXNPB) KNP=MXNPB-NPR
      DO 109 I=1,KNP
      L=NPR+I
      READ(3) NPN,NADJNP(L),NADJEL(L),(NPADJ(L,J),J=1,MXADJP)
      READ(4) NPN,R(L),Z(L),ITYPE(L),THETA(L)
  109 CONTINUE
      NPOUT=NUMCP
      GO TO 6
C
C**** TRANSFER PART OF NODE BUFFER REGION
C
  902 DC 903 K=1,NPR
      L=NUMNPB+K
      NADJNP(K)=NADJNP(L)
      NADJEL(K)=NADJEL(L)
      ITYPE(K)=ITYPE(L)
      THETA(K) =THETA(L)
      R(K)     =R(L)
      Z(K)     =Z(L)
      XMASS(K) =XMASS(L)
      SNPUU(K) =SNPUU(L)
      SNPUW(K) =SNPUW(L)
      SNPWW(K) =SNPWW(L)
      DC 903 J=1,MXADJP
      NPADJ(K,J)=NPADJ(L,J)
      SADUU(K,J)=SADJJ(L,J)
      SADUW(K,J)=SADUW(L,J)
      SADWU(K,J)=SADWJ(L,J)
  903 SADWW(K,J)=SACWW(L,J)
C
      GC TO 107
C
C**** ZERC OUT BUFFER REGION SECTION ASSOCIATED WITH NODE DATA
C
  900 DC 901 L=KX,MXNPB
      NADJNP(L)=0
      NADJEL(L)=0
      ITYPE(L)=0
      THETA(L)=0.0
      R(L)=0.0
      Z(L)=0.0
      XMASS(L)=0.0
      SNPUU(L)=0.0
      SNPUW(L)=0.0
      SNPWW(L)=0.0
      DC 901 J=1,MXADJP
      NPADJ(L,J)=0
      SADUU(L,J)=0.0
      SADUW(L,J)=0.0
      SADWU(L,J)=0.0
  901 SADWW(L,J)=0.0
C
      GO TO (2,108),ISWTCH
C
```

```
      END
      SUBROUTINE ELAST(IELAST,ISTRES,E1,E2,E3,E4,E5,C,NUME)
C
      DIMENSION C(4,4)
C
C**** FORM STRESS-STRAIN MATRIX
C
      DO 1 I=1,4
      DO 1 J=1,4
    1 C(I,J)=0.0
C
      IF(IELAST.NE.1) GO TO 20
C
C**** ISOTROPIC ELASTIC MATERIAL
C
      IF(ISTRES.EQ.2) GO TO 4
C
C     AXISYMMETRIC OR PLANE STRAIN PROBLEM
C
      EBAR=E1/((1.+E2)*(1.-2.*E2))
      C(1,1)=EBAR*(1.-E2)
      C(1,2)=EBAR*E2
      C(1,3)=C(1,2)
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=C(1,2)
      C(3,1)=C(1,2)
      C(3,2)=C(1,2)
      C(3,3)=C(1,1)
      C(4,4)=EBAR*(1.-2.*E2)/2.
      RETURN
C
C     PLANE STRESS PROBLEM
C
    4 EBAR=E1/(1.-E2*E2)
      C(1,1)=EBAR
      C(3,1)=EBAR*E2
      C(1,3)=C(3,1)
      C(3,3)=C(1,1)
      C(4,4)=EBAR*(1.-E2)/2.
      RETURN
C
C**** ANISOTROPIC ELASTIC MATERIAL
C
   20 IF(IELAST.NE.2) GO TO 30
C
      IF(ISTRES.EQ.2) GO TO 2
C
      C(1,1)=E1
      C(1,2)=E1-2.*E5
      C(1,3)=E3
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=C(1,3)
      C(3,1)=C(1,3)
      C(3,2)=C(2,3)
      C(3,3)=E2
      C(4,4)=E4
      RETURN
C
```

```
    2 C(1,1)=2.*E5*(E1-2.*E5)/E1
      C(1,3)=2.*E3*E5/E1
      C(3,1)=C(1,3)
      C(3,3)=E2-E3**2/E1
      C(4,4)=E4
      RETURN
C
   21 WRITE(6,3) IELAST,NUME,ISTRES
    3 FCRMAT(1H1/31H FRROR IN ELASTIC CONSTANT DATA/
     113H IELAST      =,I5/13H ELEMENT NO.=,I5/
     213H ISTRES      =,I5)
      CALL EXIT
C
   30 IF(IELAST.NE.3) GO TO 21
C
C**** COMPRESSIBLE FLUID
C
      IF(ISTRES.  .2) GO TO 21
C
      DC 31 I=1,3
      DC 31 J=1,3
   31 C(I,J)=E1
C
      RETURN
      END
C
C
C
      SUBROUTINE STIFF(KASE,NPI,NPJ,NPK,NPL,NUME,MAXNP,ISTRES,C,R,Z,
     1 CK,AI,S1,C1,NCRACK)
C
      DIMENSICN C(4,4),R(MAXNP),Z(MAXNP),CK(8,8),AI(23),D(8,8),G(8,8),
     1VEC(8)
C
C**** COMPUTE ELEMENT STIFFNESS MATRIX
C
C     KASE  =1 GENERAL TRIANGLE
C           =2 NCDE I ON Z-AXIS
C           =3 NCDES I,K ON Z-AXIS
C           =4 GENERAL RECTANGLE
C           =5 NCDE I ON Z-AXIS
C           =6 NCDES I,L ON Z-AXIS
C     C     =ELASTIC MODULI MATRIX
C     R     =RADIAL COORDINATE OF NODE POINTS
C     Z     =VERTICAL COORDINATE OF NODE POINTS
C     CK    =STIFFNESS MATRIX
C     AI    =INTEGRALS FOR COMPUTING K AND M
C     NUME  =ELEMENT NUMBER
C
C
      DO 1 I=1,8
      DC 1 J=1,8
      D(I,J)=0.0
      G(I,J)=0.0
    1 CK(I,J)=0.0
      IF(NCRACK.EQ.1) GO TO 300
      CALL INTER(KASE,NPI,NPJ,NPK,NPL,ISTRES,R,Z,AI,S1,C1,MAXNP)
C
      IF(NPL.NE.0) GO TO 300
C
```

```
C     TRIANGULAR ELEMENTS
C
      AJ=R(NPJ)-R(NPI)
      AK=R(NPK)-R(NPI)
      BJ=Z(NPJ)-Z(NPI)
      BK=Z(NPK)-Z(NPI)
      H=AJ*BK-AK*BJ
      B=BJ-BK
      A=AJ-AK
      IF(KASE.N..T GO TO 2
      D(1,1)=1.0
      D(2,1)=B/H
      D(3,1)=-A/H
    2 D(4,2)=1.0
      D(5,2)=B/H
      D(6,2)=-A/H
      D(2,3)=BK/H
      D(3,3)=-AK/H
      D(5,4)=D(2,3)
      D(6,4)=D(3,3)
      IF(KASE.EQ.3) GO TO 3
      D(2,5)=-BJ/H
      D(3,5)=AJ/H
    3 D(5,6)=-BJ/H
      D(6,6)=AJ/H
      NORD=6
C
      IF(KASE.NE.1) GO TO 4
      IF(ISTRES.NE.0) GO TO 4
      G(1,1)=C(2,2)*AI(5)
      G(2,1)=C(1,2)*AI(1)+C(2,2)*AI(7)
      G(3,1)=C(2,2)*AI(6)
      G(6,1)=C(2,3)*AI(1)
    4 G(2,2)=C(1,1)*AI(4)+2.*C(1,2)*AI(3)+C(2,2)*AI(10)
      IF(KASE.EQ.3) GO TO 5
      G(3,2)=C(1,2)*AI(2)+C(2,2)*AI(9)
      G(3,3)=C(2,2)*AI(8)+C(4,4)*AI(4)
      G(5,3)=C(4,4)*A (4)
      G(6,3)=C(2,3)  (2)
    5 G(6,2)=C(1,3)*AI(4)+C(2,3)*AI(3)
      G(5,5)=C(4,4)*AI(4)
      G(6,6)=C(3,3)*AI(4)
      GO TO 301
C
C     RECTANGULAR ELEMENTS
C
  300 AJ=R(NPJ)-R(NPI)
      BJ=Z(NPJ)-Z(NPI)
      A=SQRT(AJ*AJ+BJ*BJ)
      AL=R(NPL)-R(NPI)
      BL=Z(NPL)-Z(NPI)
      B=SQRT(AL*AL+BL*BL)
      H=A*B
      IF(NCRACK.EQ.0) GO TO 9
      AL=A
      A=1.0
      B=1.0
      H=1.0
    9 IF(KASE.NE.4) GO TO 6
      D(1,1)=1.0
```

```
      D(2,1)=-B/H
      D(3,1)=1./H
      D(4,1)=-A/H
    6 D(5,2)=1.0
      D(6,2)=-B/H
      D(7,2)=1./H
      D(8,2)=-A/H
      D(2,3)=-D(6,2)
      D(3,3)=-D(7,2)
      D(6,4)=D(2,3)
      D(7,4)=D(3,3)
      D(3,5)=D(7,2)
      D(7,6)=D(7,2)
      IF(KASE.EQ.6) GO TO 7
      D(3,7)=D(3,3)
      D(4,7)=-D(8,2)
    7 D(7,8)=D(7,4)
      D(8,8)=-D(8,2)
      NORD=8
      IF(NCRACK.EQ.1) GO TO 10
C
      IF(KASE.NE.4) GO TO 8
      IF(ISTRES.NE.0) GO TO 8
      G(1,1)=C(2,2)*AI(5)
      G(2,1)=C(1,2)*C1*AI(1)+C(2,2)*AI(7)
      G(3,1)=C(1,2)*(C1*AI(2)+S1*AI(3))+C(2,2)*AI(9)
      G(4,1)=C(1,2)*S1*AI(1)+C(2,2)*AI(6)
      G(6,1)=-C(2,3)*S1*AI(1)
      G(7,1)=C(2,3)*(C1*AI(3)-S1*AI(2))
      G(8,1)=C(2,3)*(C1*AI(1))
C
    8 DUM1=C1*AI(14)+S1*AI(13)
      DUM2=C1*AI(13)-S1*AI(14)
      DUM3=C1*C1*AI(12)+2.*S1*C1*AI(15)+S1*S1*AI(11)
      DUM4=S1*S1*AI(12)-2.*S1*C1*AI(15)+C1*C1*AI(11)
      G(2,2)=C1*(C(1,1)*C1*AI(4)+2.*C(1,2)*AI(3)) C(2,2)*AI(10)
     1+C(4,4)*S1*S1*AI(4)
      G(3,2)=C(1,1)*C1*DUM1+C(1,2)*(2.*C1*AI(16)+S1*AI(18))
     1+C(2,2)*AI(17)-C(4,4)*S1*DUM2
      G(7,2)=C(1,3)*C1*DUM2+C(2,3)*(C1*AI(18)-S1*AI(16))
     1-C(4,4)*S1*DUM1
      G(8,2)=AI(4)*(C(1,3)*C1*C1-C(4,4)*S1*S1)+C(2,3)*C1*AI(3)
      G(3,3)=C(1,1)*DUM3+2.*C(1,2)*(C1*AI(20)+S1*AI(23))
     1+C(2,2)*AI(21)+C(4,4)*DUM4
      G(6,3)=-C(1,3)*S1*DUM1-C(2,3)*S1*AI(16)+C(4,4)*C1*DUM2
      G(7,3)=(C(1,3)+C(4,4))*(S1*C1*(AI(11)-AI(12))+AI(15)*(C1*C1-S1*S1)
     1)+C(2,3)*(C1*AI(23)-S1*AI(20))
      G(8,3)=C(1,3)*C1*DUM1+C(2,3)*C1*AI(16)+C(4,4)*S1*DUM2
      G(6,6)=(C(3,3)*S1*S1+C(4,4)*C1*C1)*AI(4)
      G(7,6)=-C(3,3)*S1*DUM2+C(4,4)*C1*DUM1
      G(7,7)=C(3,3)*DUM4+C(4,4)*DUM3
      G(8,7)=C(3,3)*C1*DUM2+C(4,4)*S1*DUM1
      G(8,8)=(C(3,3)*C1*C1+C(4,4)*S1*S1)*AI(4)
      IF(KASE.EQ.6) GO TO 301
      G(4,2)=(C(1,1)-C(4,4))*S1*C1*AI(4)+C(1,2)*(C1*AI(2)+S1*AI(3))
     1+C(2,2)*AI(9)
      G(5,2)=-(C(1,3)+C(4,4))*S1*C1*AI(4)-C(2,3)*S1*AI(3)
      G(4,3)=C(1,1)*S1*DUM1+C(1,2)*(C1*AI(19)+2.*S1*AI(16))
     1+C(2,2)*AI(22)+C(4,4)*C1*DUM2
      G(4,4)=C(1,1)*S1*S1*AI(4)+2.*C(1,2)*S1*AI(2)+C(2,2)*AI(8)
```

```
     1+C(4,4)*C1*C1*AI(4)
      G(6,4)=AI(4)*(C(4,4)*C1*C1-C(1,3)*S1*S1)-C(2,3)*S1*AI(2)
      G(7,4)=C(1,3)*S1*DUM2+C(2,3)*(C1*AI(16)-S1*AI(19))+C(4,4)*C1*DUM1
      G(8,4)=(C(1,3)+C(4,4))*S1*C1*AI(4)+C(2,3)*C1*AI(2)
      G(8,5)=S1*C1*(C(4,4)-C(3,3))*AI(4)
C
  301 DO 201 I=2,NORD
      K=I-1
      DO 201 J=1,K
  201 G(J,I)=C(I,J)
C
C
      DO 51 J=1,NORD
      DO 50 L=1,NORD
      VEC(L)=0.0
      DO 50 K=1,NORD
   50 VEC(L)=VEC(L) G(L,K)*D(K,J)
      DO 51 I=1,NORD
      CK(I,J)=0.0
      DO 51 L=1,NORD
   51 CK(I    =CK(I,J)+D(L,I)*VEC(L)
C
      RETURN
C
   10 C1=AJ/AL
      S1=-BJ/AL
      H1=C(1,1)*S1*S1+C(4,4)*C1*C1
      H2=(C(1,3)+C(4,4))*S1*C1
      H3=C(3,3)*C1*C1+C(4,4)*S1*S1
      IF(ISTRES.EQ.0) GO TO 11
      G(3,3)=AL*H1/3.
      G(4,3)=AL*H1/2.
      G(7,3)=AL*H2/3.
      G(8,3)=AL*H2/2.
      G(4,4)=3.*G(3,3)
      G(7,4)=G(8,3)
      G(8,4)=2.*G(7,4)
      G(7,7)=AL*H3/3.
      G(8,7)=AL*H3/2.
      G(8,8)=2.*G(8,7)
      GO TO 301
   11 RI=R(NPI)
      IF(ABS(S1).GT.0.01) GO TO 12
      G(3,3)=AL*RI*H1/3.
      IF(KASE.EQ.6) GO TO 14
      G(4,3)=AL*H1*(RI/2.+AL/3.)
      G(4,4)=AL*H1*(RI+AL/2.)
   14 G(7,7)=AL*RI*H3/3.
      G(8,7)=AL*H3*(RI/2.+AL/3.)
      G(8,8)=AL*H3*(RI+AL/2.)
      GO TO 301
   12 IF(ABS(C1).GT.0.01) GO TO 13
      G(3,3)=AL*H1*(RI/3.+AL/4.)
      G(4,3)=AL*RI*H1/2.
      G(4,4)=AL*RI*H1
      G(7,7)=AL*H3*(RI/3.+AL/4.)
      G(8,7)=AL*RI*H3/2.
      G(8,8)=2.*G(8,7)
      GO TO 301
   13 RIST=RI+AL*C1
```

```
      H4=(AL**3)*(RI/3.+AL*C1/4.)
      H5=(-AL**3/(10.*S1))*(3.*RI+2.*AL*C1)-AL*AL*RI**2/(30.*S1*S1*
     1 C1)+AL*RI**3*(1.-RI/RIST)/(30.*S1*S1*C1*C1)+AL**3*(10.*RI**2+15.
     2 *AL*RI*C1+6.*AL**2*C1*C1)/(30.*RIST*S1*S1)
      H6=AL**2*(RI/2.+AL*C1/3.)
      H7=-AL**2*(5.*RI+3.*AL*C1)/(12.*S1*S1)-AL*RI**2*(1.-RI/RIST)/
     1 (12.*S1*S1*C1)+AL**2*(6.*RI**2+8.*AL*RI*C1+3.*AL**2*C1*C1)/
     2 (12.*S1*S1*RIST)
      H8=AL*(RI+AL*C1/2.)
      G(3,3)=H1*H4+C(2,2)*H5
      G(4,3)=H1*H6+C(2,2)*H7
      G(7,3)=H2*H4
      G(8,3)=H2*H6
      G(4,4)=H1*H8
      G(7,4)=H2*H6
      G(8,4)=H2*H8
      G(7,7)=H3*H4
      G(8,7)=H3*H6
      G(8,8)=H3*H8
      GO TO 301
C
      END
C
C
C
C
      SUBROUTINE INTER(KASE,NPI,NPJ,NPK,NPL,ISTRES,R,Z,AI,S1,C1,MAXNP)
C
C**** COMPUTE ELEMENT INTEGRALS
      DIMENSION AI(23),R(MAXNP),Z(MAXNP)
C
      DO 1 I=1,23
    1 AI(I)=0.
C
      IF(NPL.NE.0) GO TO 300
C
C     TRIANGULAR ELEMENTS
C
      AJ=R(NPJ)-R(NPI)
      AK=R(NPK)-R(NPI)
      BJ=Z(NPJ)-Z(NPI)
      BK=Z(NPK)-Z(NPI)
      H=AJ*BK-AK*BJ
      B=BJ-BK
      A=AJ-AK
      RI=R(NPI)
C
      IF(ISTRES.EQ.0) GO TO 2
      AI(4)=H/2.
      AI(13)=H*(AJ+AK)/6.
      AI(14)=H*(BJ+BK)/6.
      RETURN
C
    2 AI(1)=H/2.
      AI(2)=H*(BJ+BK)/6.
      AI(3)=H*(AJ+AK)/6.
      AI(4)=RI*AI(1)+AI(3)
      IF(KASE.EQ.1) GO TO 3
      AI([illegible])=AI(2)
      AI([illegible])=AI(4)
```

```
C
    3 ICOUNT=1
      RA=RI
      RB=R(NPJ)
      C=BJ/AJ
      D=0.
      DUM=-1.
      IF(C.EQ.0.) GO TO 100
  101 IF(KASE.EQ.1) GO TO 102
      IF(KASE.NE.2) GO TO 104
      IF(RA.NE.0.) GO TO 102
      FO=ALOG(RB)
      GO TO 104
  102 FO=ALOG(RB/RA)
  104 DUM1=RB-RA
      DUM2=RB*RB-RA*RA
      DUM3=RB*RB*RB-RA*RA*RA
      DUM4=DUM2*(RB*RB+RA*RA)
      IF(KASE.EQ.3)GO TO 103
      F1=DUM1-RI*FO
      F2=DUM2/2.-2.*RI*DUM1+RI*RI*FO
      F3=DUM3/3.-1.5*RI*DUM2+3.*RI*RI*DUM1-RI*RI*RI*FO
  103 GO=DUM2/2.
      G1=DUM3/3.-RI*DUM2/2.
      G2=DUM4/4.-2.*RI*DUM3/3.+RI*RI*DUM2/2.
C
      IF(KASE.NE.1) GO TO 105
      AI(5)=AI(5)+DUM*(C*F1+D*FO)
      AI(6)=AI(6)+DUM*(C*C*F2/2.+C*D*F1+D*D*FO/2.)
      AI(7)=AI(7)+DUM*(C*F2+D*F1)
      AI(9)=AI(9)+DUM*(C*C*F3/2.+C*D*F2+D*D*F1/2.)
      AI(10)=AI(10)+DUM*(C*F3+D*F2)
      GO TO 106
  105 IF(KASE.NE.2) GO TO 107
  106 AI(8)=AI(8)+DUM*(C*C*C*F3/3.+C*C*D*F2+C*D*D*F1+D*D*D*FO/3.)
  107 AI(13)=AI(13)+DUM*(C*G2+D*G1)
      AI(14)=AI(14)+DUM*(C*C*G2/2.+C*D*G1+D*D*GO/2.)
C
  100 GO TO(201,202,203), ICOUNT
  201 ICOUNT=2
      IF(A.EQ.0.) GO TO 100
      RB=R(NPJ)
      RA=R(NPK)
      C=B/A
      D=H/A
      DUM=+1.
      GO TO 101
  202 ICOUNT=3
      IF(AK.EQ.0.) GO TO 100
      IF(BK.EQ.0.) GO TO 100
      RB=R(NPK)
      RA=RI
      C=BK/AK
      D=0.
      DUM=+1.
      GO TO 101
  203 RETURN
C
C     RECTANGULAR ELEMENTS
C
```

```
  300 AJ=R(NPJ)-R(NPI)
      BJ=Z(NPJ)-Z(NPI)
      A=SQRT(AJ*AJ+BJ*BJ)
      AL=R(NPL)-R(NPI)
      BL=Z(NPL)-Z(NPI)
      B=SQRT(AL*AL+BL*BL)
      H=A*B
      S1=-BJ/A
      C1=AJ/A
      RI=R(NPI)
      IF(S1.GT..01) GO TO 301
      S1=0.
      C1=1.
      GC TO 302
  301 IF(C1.GT..01) GO TO 302
      S1=1.
      C1=0.
  302 S2=S1*S1
      S3=S2*S1
      C2=C1*C1
      C3=C2*C1
C
      IF(ISTRES.EQ.0) GO TO 303
      AI(4)=H
      AI(11)=A*A*H/3.
      AI(12)=B*B*H/3.
      AI(13)=A*H/2.
      AI(14)=B*H/2.
      AI(15)=H*H/4.
      RETURN
C
  303 AI(1)=H
      AI(2)=B*H/2.
      AI(3)=A*H/2.
      AI(4)=RI*AI(1)+C1*AI(3)+S1*AI(2)
      AI(18)=A*A*H/3.
      AI(19)=B*B*H/3.
      AI(20)=B*H*H/6.
      AI(23)=A*H*H/6.
      AI(16)=H*H/4.
      AI(15)=RI*AI(16)+C1*AI(23)+S1*AI(20)
      AI(14)=RI*AI(2)+C1*AI(16)+S1*AI(19)
      AI(13)=RI*AI(3)+C1*AI(18)+S1*AI(16)
      AI(12)=RI*AI(19)+C1*AI(20)+S1*B**3*H/4.
      AI(11)=RI*AI(18)+S1*AI(23)+C1*A**3*H/4.
      IF(KASE.NE.6) GO TO 304
      AI(2)=0.
      AI(10)=AI(4)
      AI(17)=AI(14)
      AI(19)=0.
      AI(21)=AI(12)
      RETURN
C
  304 A2=A*A
      A3=A2*A
      B2=B*B
      B3=B2*B
      RI2=RI*RI
      RI3=RI2*RI
      IF(S1.NE.0.) GO TO 305
```

```
      D1=ALOG(1.+A/RI)
      AI(5)=B*D1
      AI(6)=B2*D1/2.
      AI(7)=H-B*RI*D1
      AI(8)=B3*D1/3.
      AI(9)=B2*(A-RI*D1)/2.
      AI(10)=B*(A2/2.-A*RI+RI2*D1)
      AI(17)=B2*RI2*D1/2.-B*H*RI/2.+H*H/4.
      AI(21)=B3*(RI2*D1+A2/2.-A*RI)/3.
      AI(22)=B2*(H-B*RI*D1)/3.
      RETURN
C
  305 IF(C1.NE.0.) GO TO 306
      D1=ALOG(1.+B/RI)
      AI(5)=A*D1
      AI(6)=H-A*RI*D1
      AI(7)=A2*D1/2.
      AI(8)=A*RI2*D1-  *RI+B*H/2.
      AI(9)=A2*(B-RI*D1)/2.
      AI(10)=A3*D1/3.
      AI(17   A2*H/3.-A3*RI*D1/3.
      AI(21)=A3*(RI2*D1+B2/2.-B*RI)/3.
      AI(22)=H*H/4.-A*H*RI/2.+A2*RI2*D1/2.
      RETURN
C
  306 D1=ALOG(1.-H*S1*C1/((RI+B*S1)*(RI+A*C1)))
      D2=ALOG(1.+A*C1/(RI+B*S1))
      D3=ALOG(1.+B*S1/(RI+A*C1))
      AI(5)=(RI*D1+B*D2*S1+A*D3*C1)/(S1*C1)
      AI(6)=(B2*D2*S2-RI2*D1-A*C1*D3*(2.*RI+A*C1)+H*S1*C1)/(2.*S2*C1)
      AI(7)=(A2*D3*C2-RI2*D1-B*S1*D2*(2.*RI+B*S1)+H*S1*C1)/(2.*S1*C2)
      AI(8)=(B*H*S2*C1/6.-2.*H*RI*S1*C1/3.-A*H*S1*C2/3.+RI3*D1/3.
     1+B3*S3*D2/3.+C1*A*(RI2+A*RI*C1+A2*C2/3.)*D3)/(S3*C1)
      AI(9)=(A*H*S1*C2/3.+B*H*S2*C1/3.+H*RI*S1*C1/6.+RI3*D1/6.
     1-S2*(B2*RI/2.+B3*S1/3.)*D2-C2*(A2*RI/2.+A3*C1/3.)*D3)/(S2*C2)
      AI(10)=(A*H*S1*C2/6.-2.*H*RI*S1*C1/3.-B*H*S2*C1/3.+RI3*D1/3.
     1+A3*C3*D3/3.+S1*B*(RI2+B*RI*S1+B2*S2/3.)*D2)/(S1*C3)
      AI(17)=(A2*H*S1*C3/4.-RI*RI3*D1/12.+S2*B2*(RI2/2.+2.*B*RI*S1/3.
     1+B2*S2/4.)*D2-C3*A3*(RI/3.+A*C1/4.)*D3-S1*C1*H*(RI2/12.+5.*B*RI
     2*S1/12.+B2*S2/4.)+S1*C2*H*(A*RI/12.+H*S1/8.))/(S2*C3)
      AI(21)=-H*(A3*C1/S2+B3*S1/C2)/5.+H*H*(A/S1+B/C1)/10.-RI2*H*(B/C1
     1+A/S1)/(30.*S1*C1)-.3*RI*H*(A2/S2+B2/C2)+RI3*H/(30.*S2*C2)
     2+RI*H*H/(20.*S1*C1)+RI2*RI3*D1/(30.*S3*C3)+B3*D2*(RI2/3.+B*RI*S1
     3/2.+B2*S2/5.)/C3+A3*D3*(RI2/3.+A*RI*C1/2.+A2*C2/5.)/S3
      AI(22)=H*H/(8.*S1)-5.*A*H*RI/(12.*S2)-A2*H*C1/(4.*S2)-H*RI2/(12.*
     1S2*C1)+B*H*RI/(12.*S1*C1)+B2*H/(4.*C1)+A2*D3*(RI2/2.+2.*A*RI*C1
     2/3.+A2*C2/4.)/S3-RI*RI3*D1/(12.*S3*C2)-B3*D2*(RI/3.+B*S1/4.)/C2
C
      RETURN
C
      END
C
C
C
      SUBROUTINE ADJUSK(MAXNP,BK,ITYPE,THETA,NPI,NPJ,NPK,NPL)
C
      DIMENSION BK(8,8),BKBAR(8,8),C(8,8),VEC(8)
      DIMENSION ITYPE(MAXNP),THETA(MAXNP)
C
      NCRD=8
```

```
      IF(NPL.EQ.0) NORD=6
C
      ISWTCH=0
      IF(ITYPE(NPI).NE.1) GO TO 3
      MX=1
      NP=NPI
      ICOUNT=1
      GC TO 6
    3 IF(ITYPE(NPJ).NE.1) GO TO 4
      MX=3
      NP=NPJ
      ICOUNT=2
      GC TO 6
    4 IF(ITYPE(NPK).NE.1) GO TO 5
      MX=5
      NP=NPK
      ICOUNT=3
      GC TO 6
    5 IF(NPL.EQ.0) GO TO 7
      IF(ITYPE(NPL).NE.1) GO TO 7
      MX=7
      NP=NPL
      ICOUNT=4
C
    6 IF(ISWTCH.EQ.1) GO TO 8
      ISWTCH=1
      DC 1 I=1,NCRD
      DO 1 J=1,NCRD
      IF(I.NE.J) GO TO 2
      C(I,J)=1.0
      CC TO 1
    2 C(I,J)=0.0
    1 BKBAR(I,J)=0.0
C
    8 NX=MX+1
      C(MX,MX)=CCS(THETA(NP))
      C(NX,MX)=SIN(THETA(NP))
      C(MX,NX)=-C(NX,MX)
      C(NX,NX)=C(MX,MX)
      GO TO (3,4,5,9), ICOUNT
    7 IF(ISWTCH.EQ.0) RETURN
C
    9 DC 51 J=1,NORD
      DC 50 L=1,NORD
      VEC(L)=0.0
      DC 50 K=1,NORD
   50 VEC(L)=VEC(L)+BK(L,K)*C(K,J)
      DO 51 I=1,NORD
      BKBAR(I,J)=0.0
      DC 51 L=1,NORD
   51 BKBAR(I,J)=BKBAR(I,J)+C(L,I)*VEC(L)
C
      DC 52 I=1,NORD
      DC 52 J=1,NORD
   52 BK(I,J)=BKBAR(I,J)
C
      RETURN
      END
C
C
```

```
C
      SUBROUTINE DISTK(MAXNP,MXADJP,BK,SNPUU,SNPUW,SNPWW,SADUU,
     1SADUW,SADWU,SADWW,NPI,NPJ,NPK,NPL,NPADJ,NPOUT)
C
      DIMENSION BK(8,8),SNPUU(MAXNP),SNPUW(MAXNP),SADUU(MAXNP,MXADJP),
     1SADUW(MAXNP,MXADJP),SNPWW(MAXNP),SADWU(MAXNP,MXADJP),
     2SADWW(MAXNP,MXADJP),NPADJ(MAXNP,MXADJP)
C
C**** DISTRIBUTE ELEMENT STIFFNESS TO NODE POINT STIFFNESS
C
C     BK    =ELEMENT STIFFNESS, 6*6 FOR TRIANGLE, 8*8 FOR RECT
C     SNPUU =NODE PT. STIFF, U-DIRECTION, U-DISPL.
C     SNPUW =NODE PT. STIFF, U-DIRECTION, W-DISPL.
C     SNPWW =NODE PT. STIFF, W-DIRECTION, W-DISPL.
C     SADUU =ADJ. PT. STIFF, U-DIRECTION, U-DISPL.
C     SADUW =ADJ. PT. STIFF, U-DIRECTION, W-DISPL.
C     SADWU =ADJ. PT. STIFF, W-DIRECTION, U-DISPL.
C     SADWW =ADJ. PT. STIFF, W-DIRECTION, W-DISPL.
C
      ICOUNT=1
    1 GO TO (2,3,4,5,6),ICOUNT
C
    2 NI=NPI
      NJ=NPJ
      NK=NPK
      NL=NPL
      MX=1
      LX=3
      LY=5
      LZ=7
      GO TO 100
C
    3 NI=NPJ
      NJ=NPI
      NK=NPK
      NL=NPL
      MX=3
      LX=1
      LY=5
      LZ=7
      GO TO 100
C
    4 NI=NPK
      NJ=NPI
      NK=NPJ
      NL=NPL
      MX=5
      LX=1
      LY=3
      LZ=7
      GO TO 100
C
    5 IF(NPL.EQ.0) GO TO 6
      NI=NPL
      NJ=NPI
      NK=NPJ
      NL=NPK
      MX=7
      LX=1
      LY=3
```

```
      LZ=5
      GO TO 100
C
    6 RETURN
C
  100 SNPUU(NI)=SNPUU(NI)+BK(MX,MX)
      SNPUW(NI)=SNPUW(NI)+BK(MX,MX+1)
      SNPWW(NI)=SNPWW(NI)+BK(MX+1,MX+1)
      DO 101 I=1,MXADJP
      J=I
      IF((NPADJ(NI,I)-NPOUT).EQ.NJ) GO TO 102
  101 CONTINUE
  205 NPI=NPI+NPCUT
      NPJ=NPJ+NPCUT
      NPK=NPK+NPCUT
      IF(NPL.EQ.0) GO TO 204
      NPL=NPL+NPCUT
  204 WRITE(6,203) NPI,NPJ,NPK,NPL,NI,NJ,NK,NL,NPOUT,
     1(NPADJ(NI,I),I=1,MXADJP)
  203 FORMAT(1H1/32H ERROR IN STIFFNESS DISTRIBUTION//
     1    13H NPI        =,I5/13H NPJ          =,I5/13H NPK          =,I5/
     213H NPL          =,I5/10X,5I5/10X,8I5)
      CALL EXIT
C
  102 SADUU(NI,J)=SADUU(NI,J)+BK(MX,LX)
      SADUW(NI,J)=SADUW(NI,J)+BK(MX,LX+1)
      SADWU(NI,J)=SADWU(NI,J)+BK(MX+1,LX)
      SADWW(NI,J)=SADWW(NI,J)+BK(MX+1,LX+1)
C
      DO 103 I=1,MXADJP
      J=I
      IF((NPADJ(NI,I)-NPOUT).EQ.NK) GO TO 104
  103 CONTINUE
      GO TO 205
  104 SADUU(NI,J)=SADUU(NI,J)+BK(MX,LY)
      SADUW(NI,J)=SADUW(NI,J)+BK(MX,LY+1)
      SADWU(NI,J)=SADWU(NI,J)+BK(MX+1,LY)
      SADWW(NI,J)=SADWW(NI,J)+BK(MX+1,LY+1)
C
      IF(NL.EQ.0) GO TO 105
      DO 106 I=1,MXADJP
      J=I
      IF((NPADJ(NI,I)-NPOUT).EQ.NL) GO TO 107
  106 CONTINUE
      GO TO 205
  107 SADUU(NI,J)=SADUU(NI,J)+BK(MX,LZ)
      SADUW(NI,J)=SADUW(NI,J)+BK(MX,LZ+1)
      SADWU(NI,J)=SADWU(NI,J)+BK(MX+1,LZ)
      SADWW(NI,J)=SADWW(NI,J)+BK(MX+1,LZ+1)
  105 ICOUNT=ICOUNT+1
      GO TO 1
C
C
      END
C
C
      SUBROUTINE MASS(MAXNP,RHO,R,Z,AI,XMASS,S1,C1,NPI,NPJ,NPK,NPL,
     1ISTRES)
C
```

```
      DIMENSION R(MAXNP),Z(MAXNP),XMASS(MAXNP),AI(23)
C
C**** COMPUTE AND DISTRIBUTE MASS POINTS
C
      GRAV=386.4
C
      IF(NPL.NE.0) GO TO 2
C
C     TRIANGULAR ELEMENT
C
      AJ=R(NPJ)-R(NPI)
      AK=R(NPK)-R(NPI)
      BJ=Z(NPJ)-Z(NPI)
      BK=Z(NPK)-Z(NPI)
      H=AJ*BK-AK*BJ
      B=BJ-BK
      A=AJ-AK
C
      DUM1=RHO/H*GRAV
      AMI=DUM1*(H*AI(4)+B*AI(13)-A*AI(14))
      AMJ=DUM1*(BK*AI(13)-AK*AI(14))
      AMK=DUM1*(AJ*AI(14)-BJ*AI(13))
      GO TO 3
C
C     RECTANGULAR ELEMENT
C
    2 AJ=R(NPJ)-R(NPI)
      BJ=Z(NPJ)-Z(NPI)
      A=SQRT(AJ*AJ+BJ*BJ)
      AL=R(NPL)-R(NPI)
      BL=Z(NPL)-Z(NPI)
      B=SQRT(AL*AL+BL*BL)
C
      IF(ISTRES.EQ.0) GO TO 4
C
      AMI=RHO*AI(4)/4.*GRAV
      AMJ=AMI
      AMK=AMI
      AML=AMI
      GO TO 3
C
    4 DUM1=RHO/(1.+S1+C1)*GRAV
      AMI=DUM1*((S1+C1)*AI(4)-S1*AI(14)/B-C1*AI(13)/A)
      AMJ=DUM1*(AI(4)-(1.+C1)*AI(14)/B+C1*AI(13)/A)
      AMK=DUM1*(-AI(4)+(1.+C1)*AI(14)/B+(1.+S1)*AI(13)/A)
      AML=DUM1*(AI(4)+S1*AI(14)/B-(1.+S1)*AI(13)/A)
C
C
    3 XMASS(NPI)=XMASS(NPI)+AMI
      XMASS(NPJ)=XMASS(NPJ)+AMJ
      XMASS(NPK)=XMASS(NPK)+AMK
      IF(NPL.EQ.0) RETURN
      XMASS(NPL)=XMASS(NPL)+AML
      RETURN
C
      END
C
C
C
      SUBROUTINE PRNK(MAXNP,MXADJP,NADJNP,NPADJ,NADJEL,IPRINT,SNPUU,
```

```
     1SNPUW,SNPWW,SADUU,SADUW,SADWU,SADWW,THETA,ITYPE,XMASS,NPOUT,
     2NUMNP)
C
      DIMENSION NADJNP(MAXNP),NPADJ(MAXNP,MXADJP),NADJEL(MAXNP),
     1XMASS(MAXNP),          SNPUU(MAXNP),SNPUW(MAXNP),SNPWW(MAXNP),
     2SADUU(MAXNP,MXADJP),SADUW(MAXNP,MXADJP),SADWU(MAXNP,MXADJP),
     3SADWW(MAXNP,MXADJP),THETA(MAXNP),ITYPE(MAXNP)
C
      IF((IPRINT.NE.2).AND.(IPRINT.NE.99))GO TO 15
C
      WRITE(6,1)
    1 FORMAT(1H1,16HSTIFFNESS TABLES//6H  NODE,8X,6HSNPUU ,8X,
     16HSNPUW ,8X,6HSNPWW //)
      DO 2 I=1,NUMNP
      K=I+NPOUT
    2 WRITE(6,3) K,SNPUU(I),SNPUW(I),SNPWW(I)
    3 FORMAT(I5,3X,1P8E14.4)
C
      WRITE(6,4)
    4 FORMAT(1H1,18HADJACENT STIFFNESS//)
      WRITE(6,5)
    5 FORMAT(6H  NODE,10X,5HSADUU//)
      DO 6 I=1,NUMNP
      NUM=NADJNP(I)
      K=I+NPOUT
    6 WRITE(6,3) K,(SADUU(I,J),J=1,NUM)
C
      WRITE(6,4)
      WRITE(6,7)
    7 FORMAT(6H  NODE,10X,5HSADUW//)
      DO 8 I=1,NUMNP
      NUM=NADJNP(I)
      K=I+NPOUT
    8 WRITE(6,3) K,(SADUW(I,J),J=1,NUM)
C
      WRITE(6,4)
      WRITE(6,9)
    9 FORMAT(6H  NODE,10X,5HSADWU//)
      DO 10 I=1,NUMNP
      NUM=NADJNP(I)
      K=I+NPOUT
   10 WRITE(6,3) K,(SADWU(I,J),J=1,NUM)
C
      WRITE(6,4)
      WRITE(6,11)
   11 FORMAT(6H  NODE,10X,5HSADWW//)
      DO 12 I=1,NUMNP
      NUM=NADJNP(I)
      K=I+NPOUT
   12 WRITE(6,3) K,(SADWW(I,J),J=1,NUM)
C
   15 IF((IPRINT.NE.4).AND.(IPRINT.NE.99))RETURN
      WRITE(6,13)
   13 FORMAT(1H1,23HMASS VECTOR, LB SEC2/IN//6H  NODE//)
      DO 14 I=1,NUMNP,8
      L=I+NPC [illegible]
      NUM= [illegible]
      IF( [illegible] .GT.NUMNP) NUM=NUMNP
   14 WRITE(6,3) L,(XMASS(J),J=I,NUM)
C
```

```
      RETURN
C
C
      END
C
C
C
      OVERLAY(MOHAN,6,0      )
      PROGRAM LNK1F
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1       NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2       KTAPE,KRJN,IPRINT,NUMST,MXSTRT,IELAST(20),IPLAST(20),
     3 WGT(20),NSTART(79),ET(5,20),IPELTP,INT,NPRCDS,IMPBX
C
      DIMENSION C(4,4)
      DIMENSION NADJNP(350),NPADJ(350,8),NADJEL(350)
      DIMENSION R(350),Z(350),ITYPE(350),THETA(350)
C
      DIMENSION STNPU(4,350),STNPW(4,350),STADU(4,350,8),
     1STADW(4,350,8)
C
C**** INITIALIZE
      MOHAN=5HMOHAN
      REWIND 4
      REWIND 8
      REWIND 12
      REWIND 3
C
      KEND=0
      NPOUT=0
      NUMCP=0
      NUMNPB=0
      NPR=MXNPB
      KX=1
C
C**** ZERO OUT BUFFER REGION
C
    1 ISWTCH=1
      GO TO 900
C
C**** READ IN FIRST RECORDS
C
    2 IF(NUMNP.LT.MXNPB) NUMNPB=NUMNP
      IF(NUMNP.GE.MXNPB) NUMNPB=MXNPB
      DO 3 I=1,NUMNPB
      READ(8) NPN,NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP)
      READ(4) NPN,R(I),Z(I),ITYPE(I),THETA(I)
    3 CONTINUE
C
      ICOUNT=0
    4 READ(12) KEY,NUME,IZONE,NTI,NTJ,NTK,NTL,NCRACK,
     1((C(I,J),I=1,4),J=1,4),KASE,SI,CI
      ICOUNT=ICOUNT+1
      LNP=MAX0(NTI,NTJ,NTK,NTL)
C
      IF((LNP-NPOUT).GT.MXNPB) GO TO 100
C
C     SUFFICIENT ROOM IN BUFFER REGION
C
    6 NPI=NTI-NPOUT
```

```
      NPJ=NTJ-NPCJT
      NPK=NTK-NPCJT
      IF(NTL.EQ.0) NPL=0
      IF(NTL.NE.0) NPL=NTL-NPOUT
      CALL STRESS(MXNPB,MXADJP,C,NPI,NPJ,NPK,NPL,R,Z,KASE,NUME,NPADJ,
     1 STNPU,STNPW,STADU,STADW, ISTRES,S1,C1,NPOUT,ITYPE,THETA,NCRACK)
C
      IF(ICOUNT.LT.NUMEL) GO TO 4
      KEND=1
      KEY=NUMNP+1
C
C**** INSUFFICIENT ROOM IN BUFFER REGION
C
  100 NUMCP=KEY-1
      NUMNPB=NUMCP-NPOUT
C
C     MODIFY STRESS TABLES
  113 CALL       MODS(MXNPB,MXADJP,NADJNP,NPADJ,NADJEL,STNPU,STNPW,
     1STADU,STADW,IPRINT,THETA,ITYPE,NPOUT,NUMNPB)
C     WRITE STRESS TABLES ONTO TAPE 3
C
      DO 101 I=1,NUMNPB
  101 WRITE(3) I,NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJ[illegible](STNPU(K,I)
     1,STNPW(K,I),K=1,4),((STADU(K,I,J),STADW(K,I,J),K=1,4),J=1,MXADJP)
      IF(KEND.EQ.1) RETURN
      IF(KEND.EQ.1) RETURN
C
C     MOVE UP INCOMPLETED NODES IN BUFFER REGION
      NPR=MXNPB-NUMNPB
      GO TO 902
C     ZERO REMAINING BUFFER AREA
  107 KX=NPR+1
        WTCH=?
         TO [illegible]
C        [illegible] N REMAINING NODE POINT AND ADJACENCY DATA TO FILL IN BUFFER
C
  1   IF((NUMNP-NUMCP).LT.MXNPB) KNP=NUMNP-NUMC[illegible]NPR
      IF((NUMNP-NUMCP).GE.MXNPB) KNP=MXNPB-NPR
      DO 109 I=1,KNP
      L=NPR+I
      READ(8) NPN,NADJNP(L),NADJEL(L),(NPADJ(L,J),J=1,MXADJP)
      READ(4) NPN,R(L),Z(L),ITYPE(L),THETA(L)
  109 CONTINUE
      NPOUT=NUMCP
      GO TO 6
C**** TRANSFER PART OF NODE BUFFER REGION
C
  902 DO 903 K=1,NPR
      L=[illegible]MNPB+K
      NADJNP(K)=NADJNP(L)
      NADJEL(K)=NADJEL(L)
      ITYPE(K)=ITYPE(L)
      THETA(K)=THETA(L)
      R(K)=R(L)
      Z(K)=Z(L)
      DO 904 J=1,4
      STNPU(J,K)=STNPU(J,L)
  904 STNPW(J,K)=STNPW(J,L)
      DO 903 J=1,MXADJP
      NPADJ(K,J)=NPADJ(L,J)
```

```
      DC 903  I=1,4
      STADU(I,K,J)=STADU(I,L,J)
  903 STADW(I,K,J)=STADW(I,L,J)
C
      GO TO 107
C
C**** ZERO OUT BUFFER REGION SECTION
C
  900 DC 901  L=KX,MXNPB
      NADJNP(L)=0
      NADJEL(L)=0
      ITYPE(L)=0
      THETA(L)=0.0
      R(L)=0.0
      Z(L)=0.0
      DC 905  J=1,4
      STNPU(J,L)=0.0
  905 STNPW(J,L)=0.0
      DC 901  J=1,MXADJP
      NPADJ(L,J)=0
      DO 901  I=1,4
      STADU(I,L,J)=0.0
  901 STADW(I,L,J)=0.0
C
      GO TO (2,108),ISWTCH
C
C
      END
C
C
C
      SUBROUTINE STRESS(MAXNP,MXADJP,C,NPI,NPJ,NPK,NPL,RA,ZA,KASE,NUME,
     1NPADJ,STNPU,STNPW,STADU,STADW,ISTRES,S1,C1,NPOUT,ITYPE,THETA,
     2NCRACK)
C
      DIMENSICN C(4,4),RA(MAXNP),ZA(MAXNP),NPADJ(MAXNP,MXADJP),S(4,8),
     1STNPU(4,MAXNP),STNPW(4,MAXNP),STADU(4,MAXNP,MXADJP),
     2STADW(4,MAXNP,MXADJP),ITYPE(MAXNP),THETA(MAXNP)
C
      DIMENSICN SBAR(4,8),CBAR(8,8)
C
C**** COMPUTE NODE POINT STRESS-DISPLACEMENT RELATIONS
C
C
C     C      =STRESS-STRAIN MATRIX
C     S      =STRESS-DISPLACEMENT MATRIX FOR ELEMENT
C     ST(1) =RADIAL      STRESS, SIGMA R
C     ST(2) =MERIDIONAL STRESS, SIGMA THETA
C     ST(3) =VERTICAL    STRESS, SIGMA Z
C     ST(4) =SHEAR       STRESS, TAU
C
C     ST(I) =STNPU(I)*U+STNPW(I)*W+SUM(STADU(I)*U+STADW(I)*W)
C
      DO 1  I=1,4
      DO 1  J=1,8
    1 S(I,J)=0.0
C
      RR=RA(NPI)
      R=0.
      RP=0.
```

```
      Z=0.
      ZP=0.
      ICOUNT=1
C
      IF(NPL.NE.0) GO TO 200
C
C     TRIANGULAR ELEMENTS
C
      AJ=RA(NPJ)-RA(NPI)
      AK=RA(NPK)-RA(NPI)
      BJ=ZA(NPJ)-ZA(NPI)
      BK=ZA(NPK)-ZA(NPI)
      H=AJ*BK-AK*BJ
      B=BJ-BK
      A=AJ-AK
C
   25 IF(KASE.NE.1) GO TO 26
      IF(ISTRES.NE.0) GO TO 27
      AOR=1./RR
      ROR=R/RR
      ZOR=Z/RR
      GO TO 28
   27 IF(ICOUNT.NE.1) GO TO 100
      AOR=0.0
      ROR=0.0
      ZOR=0.0
   28 DUM1=B/H
      DUM2=AOR+(B*ROR-A*ZOR)/H
      DO 35 I=1,3
   35 S(I,1)=C(1,I)*DUM1+C(I,2)*DUM2
      S(4,1)=-C(4,4)*A/H
      GO TO 29
   26 ROR=1.0
      IF(KASE.NE.2) GO TO 30
      IF(ICOUNT.NE.1) GO TO 31
      ZOR=0.0
      GO TO 29
   31 ZOR=Z/RR
      GO TO 29
   30 ZOR=0.0
      IF(ICOUNT.NE.1) GO TO 100
   29 DO 36 I=1,3
   36 S(I,2)=-C(I,3)*A/H
      S(4,2)=C(4,4)*B/H
      DUM1=BK/H
      DUM2=(BK*ROR-AK*ZOR)/H
      DO 37 I=1,3
   37 S(I,3)=C(1,I)*DUM1+C(I,2)*DUM2
      DUM1=AK/H
      S(4,3)=-C(4,4)*DUM1
      DO 38 I=1,3
   38 S(I,4)=-C(I,3)*DUM1
      S(4,4)=C(4,4)*BK/H
      DUM1=AJ/H
      DO 39 I=1,3
   39 S(I,6)=C(I,3)*DUM1
      S(4,6)=-C(4,4)*BJ/H
      IF(KASE.EQ.3) GO TO 300
      DUM1=-BJ/H
      DUM2=(AJ*ZOR-BJ*ROR)/H
```

```
      DO 40 I=1,3
   40 S(I,5)=C(1,I)*DUM1+C(I,2)*DUM2
      S(4,5)=C(4,4)*AJ/H
      GO TO 300
C
C     RECTANGULAR ELEMENT
C
  200 AJ=RA(NPJ)-RA(NPI)
      BJ=ZA(NPJ)-ZA(NPI)
      A=SQRT(AJ*AJ+BJ*BJ)
      AL=RA(NPL)-RA(NPI)
      BL=ZA(NPL)-ZA(NPI)
      B=SQRT(AL*AL+BL*BL)
      H=A*B
C
   50 IF(KASE.NE.4) GO TO 51
      IF(ISTRES.NE.0) GO TO 52
      AOR=1./RR
      ROR=RP/RR
      ZOR=ZP/RR
      IF(NCRACK.EQ.0) GO TO 53
      H=A
      AOR=0.0
      B=0.0
      GO TO 53
   52 AOR=0.0
      ROR=0.0
      ZOR=0.0
      IF(NCRACK.EQ.0) GO TO 53
      H=A
      B=0.0
   53 DUM1=(RP*S1+ZP*C1-A*S1-B*C1)/H
      DUM2=AOR+(ROR*ZP-B*ROR-A*ZOR)/H
      DO 65 I=1,3
   65 S(I,1)=C(1,I)*DUM1+C(I,2)*DUM2
      S(4,1)=C(4,4)*(B*S1-A*C1+RP*C1-ZP*S1)/H
      GO TO 54
   51 IF(KASE.EQ.5) GO TO 73
      ROR=1.
      ZOR=0.
      IF(NCRACK.EQ.0) GO TO 54
      H=A
      AOR=0.0
      B=0.0
      GO TO 54
   73 IF(ICOUNT.NE.1) GO TO 74
      ROR=C1
      ZOR=S1
      GO TO 54
   74 ROR=RP/RR
      ZOR=ZP/RR
   54 DUM1=(B*S1-A*C1+RP*C1-ZP*S1)/H
      DO 66 I=1,3
   66 S(I,2)=C(I,3)*DUM1
      S(4,2)=C(4,4)*(RP*S1+ZP*C1-B*C1-A*S1)/H
      DUM1=(B*C1-RP*S1-ZP*C1)/H
      DUM2=(B*ROR-ROR*ZP)/H
      DO 67 I=1,3
   67 S(I,3)=C(1,I)*DUM1+C(I,2)*DUM2
      DUM1=(-B*S1-RP*C1+ZP*S1)/H
```

```
      S(4,3)=C(4,4)*DJM1
      DO 68 I=1,3
   68 S(I,4)=C(I,3)*DJM1
      S(4,4)=C(4,4)*(B*C1-RP*S1-ZP*C1)/H
      DUM1=(RP*S1+ZP*C1)/H
      DUM2=ROR*ZP/H
      DO 69 I=1,3
   69 S(I,5)=C(1,I)*DJM1+C(I,2)*DUM2
      DUM1=(RP*C1-ZP*S1)/H
      S(4,5)=C(4,4)*DJM1
      DO 70 I=1,3
   70 S(I,6)=C(I,3)*DJM1
      S(4,6)=C(4,4)*(ZP*C1+RP*S1)/H
      DUM1=(A*C1-RP*C1+ZP*S1)/H
      DO 71 I=1,3
   71 S(I,8)=C(I,3)*DJM1
      S(4,8)=C(4,4)*(A*S1-RP*S1-ZP*C1)/H
      IF(KASE.EQ.6) GO TO 300
      S(4,7)=C(4,4)*DJM1
      DUM1=(A*S1-RP*S1-ZP*C1)/H
      DUM2=(A*ZOR-ROR*ZP)/H
      DO 72 I=1,3
   72 S(I,7)=C(1,I)*DJM1+C(I,2)*DUM2
C
C     MODIFY S-MATRIX
C
  300 ISWTCH=0
      IF(ITYPE(NPI).NE.1) GO TO 303
      MX=1
      NP=NPI
      KCOUNT=1
      GO TO 306
  303 IF(ITYPE(NPJ).NE.1) GO TO 304
      MX=3
      NP=NPJ
      KCOUNT=2
      GO TO 306
  304 IF(ITYPE(NPK).NE.1) GO TO 305
      MX=5
      NP=NPK
      KCOUNT=3
      GO TO 306
  305 IF(NPL.EQ.0) GO TO 307
      IF(ITYPE(NPL).NE.1) GO TO 307
      MX=7
      NP=NPL
      KCOUNT=4
C
  306 IF(ISWTCH.EQ.1) GO TO 308
      ISWTCH=1
      DO 301 I=1,8
      DO 301 J=1,8
      IF(I.NE.J) GO TO 302
      CBAR(I,J)=1.0
      GO TO 301
  302 CBAR(I,J)=0.0
  301 CONTINUE
      DO 312 I=1,4
      DO 312 J=1,8
  312 SBAR(I,J)=0.0
```

```
C
  308 NX=MX+1
      CBAR(MX,MX)=COS(THETA(NP))
      CBAR(NX,MX)=SIN(THETA(NP))
      CBAR(MX,NX)=-CBAR(NX,MX)
      CBAR(NX,NX)=CBAR(MX,MX)
      GO TO (303,304,305,309),KCOUNT
  307 IF(ISWTCH.EQ.0) GO TO 100
C
  309 DO 310 I=1,4
      DO 310 J=1,8
      SBAR(I,J)=0.0
      DO 310 K=1,8
  310 SBAR(I,J)=SBAR(I,J)+S(I,K)*CBAR(K,J)
C
      DO 311 I=1,4
      DO 311 J=1,8
  311 S(I,J)=SBAR(I,J)
C
C     DISTRIBUTE S-MATRIX
C
  100 IF(ICOUNT.NE.1) GO TO 101
      MI=1
      MJ=3
      MK=5
      ML=7
      NI=NPI
      NJ=NPJ
      NK=NPK
      NL=NPL
      GO TO 103
C
  101 IF(ICOUNT.NE.2) GO TO 102
      MI=3
      MJ=1
      MK=5
      ML=7
      NI=NPJ
      NJ=NPI
      NK=NPK
      NL=NPL
      GO TO 103
C
  102 IF(ICOUNT.NE.3) GO TO 107
      MI=5
      MJ=1
      MK=3
      ML=7
      NI=NPK
      NJ=NPI
      NK=NPJ
      NL=NPL
      GO TO 103
  107 MI=7
      MJ=1
      MK=3
      ML=5
      NI=NPL
      NJ=NPI
      NK=NPJ
```

```
      NL=NPK
C
  103 DO 114 I=1,4
      STNPU(I,NI)=STNPJ(I,NI)+S(I,MI)
  114 STNPW(I,NI)=STNPW(I,NI)+S(I,MI+1)
      N=1
  155 GC TO (150,151,152,153),N
  150 NN=NJ
      MN=MJ
      GO TO 154
  151 NN=NK
      MN=MK
      GO TO 154
  152 IF(NPL.EQ.0) GO TO 153
      NN=NL
      MN=ML
C
  154 DO 104 K=1,MXADJP
      J=K
      IF((NPACJ(NI,K)-NPOUT).EQ.NN) GO TO 105
  104 CONTINUE
  109 WRITE(6,106) NUME,NI,N,NPOUT
  106 FORMAT(1H1/30H ERROR IN STRSS, STATEMENT 104/
     113H ELEMENT NO.=,I5/13H NODE POINT =,I5/13H N            =,I5/
     213H NPOUT        =,I5)
      CALL EXIT
  105 DO 115 I=1,4
      STADU(I,NI,J)=STADJ(I,NI,J)+S(I,MN)
  115 STADW(I,NI,J)=STACW(I,NI,J)+S(I,MN+1)
      N=N+1
      GC TO 155
C
  153 GC TO (110,111,112,113),ICOUNT
  110 ICOUNT=2
      RR=RA(NPJ)
      R=AJ
      Z=BJ
      RP=A
      ZP=0.0
      GO TO 120
  111 ICOUNT=3
      RR=RA(NPK)
      R=AK
      Z=BK
      RP=A
      ZP=B
  120 IF(NPL.EQ.0) GO TO 25
      GO TO 50
  112 IF(NPL.EQ.0) RETURN
      ICOUNT=4
      RR=RA(NPL)
      RP=0.0
      ZP=B
      GC TO 50
  113 RETURN
      END
C
C
C
      SUBROUTINE MODS(MAXNP,MXADJP,NADJNP,NPADJ,NADJEL,STNPU,STNPW,
```

```
 1STADU,STADW,IPRINT,THETA,ITYPE,NPOUT,NUMNP)
  DIMENSION NADJNP(MAXNP),NPADJ(MAXNP,MXADJP),NADJEL(MAXNP),
 1STNPU(4,MAXNP),STNPW(4,MAXNP),STADU(4,MAXNP,MXADJP),
 2STADW(4,MAXNP,MXADJP),THETA(MAXNP),ITYPE(MAXNP)
  DO 2 I=1,NUMNP
  DUM=NADJEL(I)
  DO 1 K=1,4
  STNPU(K,I)=STNPU(K,I)/DUM
1 STNPW(K,I)=STNPW(K,I)/DUM
  NUM=NADJNP(I)
  DO 2 J=1,NUM
  DO 2 K=1,4
  STADU(K,I,J)=STADU(K,I,J)/DUM
2 STADW(K,I,J)=STADW(K,I,J)/DUM
  IF((IPRINT.NE.3).AND.(IPRINT.NE.99))RETURN
  WRITE(6,3)
3 FORMAT(1H1,13HSTRESS TABLES//6H  NODE,8X,6HSTRNPU,8X,6HSTRNPW,
 18X,6HSTTNPU,8X,6HSTTNPW,8X,6HSTZNPU,8X,6HSTZNPW,8X,6HSTSNPU,8X,
 26HSTSNPW//)
  DO 4 I=1,NUMNP
  L=I+NPOUT
4 WRITE(6,5) L,(STNPU(K,I),STNPW(K,I),K=1,4)
5 FORMAT(I5,3X,1P8E14.4)
  WRITE(6,6)
6 FORMAT(1H1,22HADJACENT STRESS TABLES//6H  NODE,10X,6HSTRADU/
 116X,6HSTRADW /16X,6HSTTADU/16X,6HSTTADW/16X,6HSTZADU/
 216X,6HSTZADW/16X,6HSTSADU/16X,6HSTSADW//)
  DO 7 I=1,NUMNP
  L=I+NPOUT
  WRITE(6,5) L
  NUM=NADJNP(I)
  DO 9 K=1,4
  WRITE(6,8) (STADU(K,I,J),J=1,NUM)
9 WRITE(6,8) (STADW(K,I,J),J=1,NUM)
8 FORMAT(8X,1P8E14.4)
7 CONTINUE
C
  RETURN
C
C
  END
C
C
C
  OVERLAY(MOHAN,7,0      )
  PROGRAM LNKIG
  COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPEL3,NUMNP,
 1        NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
 2        KTAPE,KRJN,IPRINT,NUMST,MXSTRT,IELAST(20),IPLAST(20),
 3 WGT(20),NSTART(79),EI(5,20),IPELTP,INT,NPRCDS,IMPBX
  DIMENSION FAU(1600),FAW(1600),R(1600),Z(1600),ITYPE(1600),
 1THETA(1600),NPTP(1600),ANAME(18),CPRESS(100,3),PRESSU(100),
 2PRESSW(100),NPLOAD(100),U(1600),W(1600),NPADJ(8),SADUU(8),
 3SADUW(8),SADWU(8),SADWW(8),PLOADU(100),PLOADW(100),UDISP(100),
 4WDISP(100),NPDISP(100)
  EQUIVALENCE(U,R),(W,Z),(PLOADU,PRESSU,UDISP),
 1          (PLOADW,PRESSW,WDISP), (NPDISP,NPLOAD)
  MOHAN=5HMOHAN
  REWIND 8
  DO 6 I=1,NUMNP
```

```
    6 READ(8) J,R(1),R(1),(R(K),K=1,MXADJP)
      READ(8) NJ,(NPLOAD(I),NPLOAD(I),NPLOAD(I),NPLOAD(I),NPLOAD(I),
     1NPLOAD(I),I=1,NJ)
      READ(8) (NU,NPTP(I),I=1,NUMNP)
      REWIND 8
      WRITE(6,2)
    2 FORMAT(1H1,21HAPPLIED PRESSURE DATA//)
      READ(5,3) NLINES
    3 FORMAT(14I5)
      WRITE(6,4) NLINES
    4 FORMAT(26H NO. OF PRESSURE SURFACES=,I5//)
      REWIND 4
      DO 5 I=1,NUMNP
      FAU(I)=0.0
      FAW(I)=0.0
    5 READ(4) N,R(I),Z(I),ITYPE(I),THETA(I)
      REWIND 4
C
      IF(NLINES.EQ.0) GO TO 20
      DO 19 ILINE=1,NLINES
      READ(5,8) LOADNP,ANAME
    8 FORMAT(I5,18A4)
      WRITE(6,9) ANAME,LOADNP,ILINE
    9 FORMAT(//18A4/20H NO. OF NODE POINTS=,I5,12H  ON SURFACE,I5//)
      WRITE(6,11)
   11 FORMAT(5H NODE,10X,6HPRESSU,14X,6HPRESSW/16X,3HPSI,17X,3HPSI//)
      DO 7 I=1,LOADNP
      READ(5,10) NPLOAD(I),PRESSU(I),PRESSW(I)
   10 FORMAT(I5,2E10.0)
    7 WRITE(6,12)  NPLOAD(I),PRESSU(I),PRESSW(I)
   12 FORMAT(I5,1P2E20.5)
      DO 13 I=1,LOADNP
      NP=NPLOAD(I)
   13 NPLOAD(I)=NPTP(NP)
      CALL COEF(MAXNP,NUMNP,LOADNP,NPLOAD,CPRESS,R,Z,ISTRES)
      DO 14 I=1,LOADNP
C**********************************************************************
      IF(I.EQ.1) GO TO 16
      DUMU=PRESSU(I-1)*CPRESS(I,1)
      DUMW=PRESSW(I-1)*CPRESS(I,1)
      GO TO 15
   16 DUMU=0.0
      DUMW=0.0
   15 DUMU=DUMU+PRESSU(I)*CPRESS(I,2)
      DUMW=DUMW+PRESSW(I)*CPRESS(I,2)
      IF(I.EQ.LOADNP) GO TO 17
      DUMU=DUMU+PRESSU(I+1)*CPRESS(I,3)
      DUMW=DUMW+PRESSW(I+1)*CPRESS(I,3)
C**********************************************************************
   17 NP=NPLOAD(I)
      FAU(NP)=FAU(NP)+DUMU
      FAW(NP)=FAW(NP)+DUMW
   14 CONTINUE
   19 CONTINUE
C
   20 WRITE(6,21)
   21 FORMAT(1H1,22HCONCENTRATED LOAD DATA//)
      READ(5,3) NLINES
      WRITE(6,22) NLINES
   22 FORMAT(22H NO. OF LOAD CLUSTERS=,I5//)
```

```
C
      IF(NLINES.EQ.0) GO TO 26
C
      DO 28 ILINE=1,NLINES
      READ(5,8) LOADNP,ANAME
      WRITE(6,23) ANAME,LOADNP,ILINE
   23 FORMAT(//18A4/14H NO. OF NODES=,I5,21H  IN LOAD CLUSTER NO.,I5//)
      WRITE(6,24)
   24 FORMAT(5H NODE,10X,5HLOADU,15X,5HLOADW/1 ,3HLBS,17X,3HLBS//)
      DO 18 I=1,LOADNP
      READ(5,10) NPLOAD(I), PLOADU(I),PLOADW(I)
 18   WRITE(6,12) NPLOAD(I), PLOADU(I), PLOADW(I)
C
      D  25 I=1,LOADNP
      NP=NPLOAD(I)
      NPLOAD(I)=NPTP(NP)
      NP=NPLOAD(I)
      FAU(NP) = FAU(NP) + PLOADU(I)
 25   FAW(NP) = FAW(NP) + PLOADW(I)
C
   28 CONTINUE
   26 DO 27 I=1,NUMNP
      IF(ITYPE(I).NE.1) GO TO 27
      DUMU=FAU(I)*COS(THETA(I))+FAW(I)*SIN(THETA(I))
      DUMW=-FAU(I)*SIN(THETA(I))+FAW(I)*COS(THETA(I))
      FAU(I)=DUMU
      FAW(I)=DUMW
   27 CONTINUE
C
      DO 29 I=1,NUMNP
      U(I)=0.0
   29 W(I)=0.0
C
      WRITE(6,30)
   30 FORMAT(1H1,17HDISPLACEMENT DATA//)
      READ(5,3) NLINES
      WRITE(6,31) NLINES
   31 FORMAT(30H NO. OF DISPLACEMENT CLUSTERS=,I5//)
C
      IF(NLINES.EQ.0) GO TO 37
      DO 32 ILINE=1,NLINES
      READ(5,8) LDISP, ANAME
      WRITE(6,33) ANAME, LDISP, ILINE
   33 FORMAT(//18A4/20H NO. OF NODE POINTS=,I5,11H ON CLUSTER,I5//)
      WRITE(6,34)
   34 FORMAT(5H NODE,10X,6HUDISPL,14X,6HWDISPL/17X,3HIN.,17X,3HIN.//)
      DO 35 I=1,LDISP
      READ(5,10)NPDISP(I),UDISP(I),WDISP(I)
 35   WRITE (6,12)NPDISP(I), UDISP(I), WDISP(I)
      DO 36 I=1,LDISP
      NP = NPDISP(I)
      NPDISP(I) = NPTP(NP)
      NP=NPDISP(I)
      U(NP)=UDISP(I)
 36   W(NP)=WDISP(I)
   32 CONTINUE
C
   37 REWIND 13
      REWIND 14
      DO 38 I=1,NUMNP
```

```
      READ(10) N,NADJNP,ITYPE(I),THETA(I),XMASS,SNPLU,SNPUW,SNPWW,
     1(NPADJ(J),SADUJ(J),SADJW(J),SADWU(J),SADWW(J),J=1,MXADJP)
      FAW(I)=FAW(I)+XMASS
   38 WRITE(14)I,NADJNP,ITYPE(I),THETA(I),XMASS,SNPLU,SNPUW,SNPWW,
     1(NPADJ(J),SADUJ(J),SADJW(J),SADWU(J),SADWW(J),J=1,MXADJP),
     2FAU(I),FAW(I),J(I),W(I)
      REWIND 10
      REWIND 14
      IF((IPRINT.NE.5).AND.(IPRINT.NE.99))RETURN
      WRITE(6,101)
  101 FORMAT(1H1,32HINPUT LOAD AND DISPLACEMENT DATA//
     15X,3HNEW,12X,6HJDISPL,14X,6HWDISPL,14X,5HLOADU,15X,5HLOADW/
     25X,4HNODE,12X,4H(IN),16X,4H(IN),15X,5H(LBS),15X,5H(LBS)//)
      DO 102 I=1,NJMNP
  102 WRITE(6,103) I,J(I),W(I),FAU(I),FAW(I)
  103 FORMAT(I8,2X,1P4E20.5)
      RETURN
C
C
C
      END
      SUBROUTINE COEF(MAXNP,NUMNP,LOADNP,NPLOAD,CPRESS,R,Z,ISTRES)
      DIMENSION NPLOAD(LOADNP),CPRESS(100,3),R(MAXNP),Z(MAXNP)
      DO 1 I=1,LOADNP
      DO 1 J=1,3
    1 CPRESS(I,J)=0.0
C
      NUM=LOADNP-1
      DO 4 I=1,NUM
      NP=NPLOAD(I)
      NP1=NPLOAD(I+1)
      AJ=R(NP1)-R(NP)
      BJ=Z(NP1)-Z(NP)
      AL=SQRT(AJ*AJ+BJ*BJ)
      IF(ISTRES.NE.0) GO TO 2
      C1=AL*(3.*R(NP)+R(NP1))/12.
      C2=AL*(R(NP)+R(NP1))/12.
      C3=AL*(R(NP)+3.*R(NP1))/12.
      GO TO 3
    2 C1=AL/3.
      C2=AL/6.
      C3=C1
    3 CPRESS(I,2)=CPRESS(I,2)+C1
      CPRESS(I,3)=C2
      M=I+1
      CPRESS(M,2)=C3
      CPRESS(M,1)=C2
    4 CONTINUE
      RETURN
      END
C
C
C
      OVERLAY(MOHAN,10,0      )
      PROGRAM LNK1H
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPEL3,NUMNP,
     1        NUMEL,ISTRES,NUMPFL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2        KTAPE,KRJN,IPRINT,NUMST,MXSTRT,IELAST(20),  LAST(20),
     3      WGT(20),NSTART(79),EI(5,20),IPELTP,INT,NPRCDS
      DIMENSION NADJNP(400),ITYPE(400),SNPUU(400),SNPUW(400),
```

```
     1SNPWU(400),SNPWW(400),NPADJ(400,16),SADUU(400,16),
     2SADUW(400,16),SADWU(400,16),SADWW(400,16),FAU(400),
     3FAW(400),U(400),W(400)
      DIMENSION UN(1600),WN(1600)
      MOHAN=5HMOHAN
C**********************************************************************
      WRITE(6,1000)
 1000 FORMAT(* ELIMINATION SOLUTION HAS STARTED*)
C**********************************************************************
      REWIND 13
      REWIND 14
      NXADJP=2*MXADJP
      NPOUT=0
      NUMCP=0
      KX=1
      GO TO 900
    1 DO 50 N=1,NUMNPB
      IDUM=NADJNP(N)
      IF(IDUM.EQ.0) GO TO 60
      DO 51 J=1,IDUM
      IF(NPADJ(N,J).GT.(NPOUT+NUMNPB)) GO TO 59
   51 CONTINUE
C
   60 IF(ITYPE(N).EQ.0) GO TO 50
      IF(IDUM.EQ.0) GO TO 61
      DO 52 J=1,IDUM
      NADJ=NPADJ(N,J)
      NRDJ=NADJ-NPOUT
      JDUM=NADJNP(NRDJ)
      DO 53 K=1,JDUM
      KK=K
      IF(NPADJ(NRDJ,K).EQ.(N+NPOUT)) GO TO 55
   53 CONTINUE
      WRITE(6,54)
   54 FORMAT(1H1,26HERROR IN LIF, STATEMENT 54//)
      NODE=N
      GO TO 909
   55 IF(ITYPE(N).EQ.1) GO TO 56
      FAU(NRDJ)=FAU(NRDJ)-SADUU(NRDJ,KK)*U(N)
      FAW(NRDJ)=FAW(NRDJ)-SADWU(NRDJ,KK)*U(N)
      SADUU(NRDJ,KK)=0.0
      SADWU(NRDJ,KK)=0.0
      SADUU(N,J)=0.0
      SADUW(N,J)=0.0
   56 IF(ITYPE(N).EQ.3) GO TO 52
      FAU(NRDJ)=FAU(NRDJ)-SADUW(NRDJ,KK)*W(N)
      FAW(NRDJ)=FAW(NRDJ)-SADWW(NRDJ,KK)*W(N)
      SADUW(NRDJ,KK)=0.0
      SADWW(NRDJ,KK)=0.0
      SADWU(N,J)=0.0
      SADWW(N,J)=0.0
   52 CONTINUE
   61 IF(ITYPE(N).GT.1) GO TO 57
      FAW(N)=W(N)
      FAU(N)=FAU(N)-SNPUW(N)*W(N)
      SNPWW(N)=1.0
      SNPUW(N)=0.0
      SNPWU(N)=0.0
      GO TO 50
   57 IF(ITYPE(N).GT.2) GO TO 58
```

```
      FAU(N)=U(N)
      FAW(N)=W(N)
      SNPUU(N)=1.0
      SNPWW(N)=1.0
      SNPUW(N)=0.0
      SNPWU(N)=0.0
      GC TO 50
   58 FAU(N)=U(N)
      FAW(N)=FAW(N)-SNPWJ(N)*U(N)
      SNPUU(N)=1.0
      SNPUW(N)=0.0
      SNPWU(N)=0.0
   50 CCNTINUE
   59 DC 3 N=1,NUMNPB
      NN=N
      JDUM=N+NPOUT
      IF(JDJM.EQ.NJMNP) GO TO 14
      IDUM=NADJNP(N)
      DC 4 J=1,ICJM
      IF(NPADJ(N,J).GT.(NPOUT+NUMNPB)) GO TO 43
    4 CONTINUE
      NRMCP=NRMCP+1
      DC 5 J=1,ICUM
      NADJ=NPADJ(N,J)
      NRDJ=NADJ-NPOUT
      DC 6 K=J,ICJM
      IF(NPADJ(N,K).EQ.NADJ) GO TO 6
      DO 7 L=1,NXADJP
      LL=L
      LA=LL
      IF(NPADJ(NRDJ,L).EQ.NPADJ(N,K)) GO TO 6
      IF(NPADJ(NRDJ,L).EQ.0) GO TO 9
    7 CONTINUE
      IF(NADJ.LT.300) GO TO 6
      WRITE(6,8)
    8 FORMAT(1H1,25HERROR IN L1F, STATEMENT 8//)
      NCDE=N
      GC TO 909
    9 NPADJ(NRDJ,LL)=NPADJ(N,K)
      NADJNP(NRDJ)=NADJNP(NRDJ)+1
      NBDJ=NPADJ(N,K)-NPOUT
      DC 10 L=1,NXADJP
      LL=L
      IF(NPADJ(NBDJ,L).EQ.0) GO TO 12
   10 CONTINJE
      NPADJ(NRDJ,LA)=0
      NADJNP(NRDJ)=NADJNP(NRDJ)-1
      IF(NADJ.LE.300) GO TO 6
      WRITE(6,11)
   11 FORMAT(1H1,26HERROR IN L1F, STATEMENT 11//)
      NCDE=N
      GO TO 909
   12 NPADJ(NBDJ,LL)=NADJ
      NADJNP(NBDJ)=NADJNP(NBDJ)+1
    6 CONTINUE
    5 CONTINUE
   29 DO 20 J=1,IDJM
      NADJ=NPADJ(N,J)
      NRDJ=NACJ-NPOUT
      JDUM=NACJNP(NRDJ)
```

```
      DO 21 K=1,JDUM
      KK=K
      IF(NPADJ(NRDJ,K).EQ.(N+NPOUT)) GO TO 23
   21 CONTINUE
      IF(NADJ.LE.300) GO TO 20
      WRITE(6,22)
   22 FORMAT(1H1,26HERROR IN L1F, STATEMENT 22//)
      NODE=N
      GO TO 909
   23 SMULU=SADUU(NRDJ,KK)
      SMULW=SADWU(NRDJ,KK)
      SNPUU(NRDJ)=SNPUU(NRDJ)-SMULU*SADUU(N,J)/SNPUU(N)
      SNPUW(NRDJ)=SNPUW(NRDJ)-SMULU*SADUW(N,J)/SNPUU(N)
      SNPWU(NRDJ)=SNPWU(NRDJ)-SMULW*SADUU(N,J)/SNPUU(N)
      SNPWW(NRDJ)=SNPWW(NRDJ)-SMULW*SADUW(N,J)/SNPUU(N)
      DO 24 K=1,JDUM
      IF(NPADJ(NRDJ,K).NE.(N+NPOUT)) GO TO 25
      SADUW(NRDJ,K)=SADUW(NRDJ,K)-SMULU*SNPUW(N)/SNPUU(N)
      SADWW(NRDJ,K)=SADWW(NRDJ,K)-SMULW*SNPUW(N)/SNPUU(N)
      SADUU(NRDJ,K)=0.0
      SADWU(NRDJ,K)=0.0
      GO TO 24
   25 DO 26 L=1,IDUM
      LL=L
      IF(NPADJ(N,L).EQ.NPADJ(NRDJ,K)) GO TO 27
   26 CONTINUE
      GO TO 24
   27 SADUU(NRDJ,K)=SADUU(NRDJ,K)-SMULU*SADUU(N,LL)/SNPUU(N)
      SADUW(NRDJ,K)=SADUW(NRDJ,K)-SMULU*SADUW(N,LL)/SNPUU(N)
      SADWU(NRDJ,K)=SADWU(NRDJ,K)-SMULW*SADUU(N,LL)/SNPUU(N)
      SADWW(NRDJ,K)=SADWW(NRDJ,K)-SMULW*SADUW(N,LL)/SNPUU(N)
   24 CONTINUE
      FAU(NRDJ)=FAU(NRDJ)-SMULU*FAU(N)/SNPUU(N)
      FAW(NRDJ)=FAW(NRDJ)-SMULW*FAU(N)/SNPUU(N)
      SADWU(N,J)=SADWU(N,J)-SNPWU(N)*SADUU(N,J)/SNPUU(N)
      SADWW(N,J)=SADWW(N,J)-SNPWU(N)*SADUW(N,J)/SNPUU(N)
   20 CONTINUE
C
   14 FAW(N)=FAW(N)-SNPWU(N)*FAU(N)/SNPUU(N)
      SNPWW(N)=SNPWW(N)-SNPWU(N)*SNPUW(N)/SNPUU(N)
      SNPUW(N)=SNPUW(N)/SNPUU(N)
      FAU(N)=FAU(N)/SNPUU(N)
      JDUM=N+NPOUT
      IF(JDUM.EQ.NUMNP) GO TO 15
      DO 28 J=1,IDUM
      SADUU(N,J)=SADUU(N,J)/SNPUU(N)
   28 SADUW(N,J)=SADUW(N,J)/SNPUU(N)
C
   30 DO 31 J=1,IDUM
      NADJ=NPADJ(N,J)
      NRDJ=NADJ-NPOUT
      JDUM=NADJNP(NRDJ)
      DO 32 K=1,JDUM
      KK=K
      IF(NPADJ(NRDJ,K).EQ.(N+NPOUT)) GO TO 34
   32 CONTINUE
      IF(NADJ.LE.300) GO TO 31
      WRITE(6,33)
   33 FORMAT(1H1,26HERROR IN L1F, STATEMENT 33//)
      NODE=N
```

```
      GO TO 909
   34 SMULU=SADUW(NRDJ,KK)
      SMULW=SADWW(NRDJ,KK)
      SNPUU(NRDJ)=SNPUU(NRDJ)-SMULU*SADWU(N,J)/SNPWW(N)
      SNPUW(NRDJ)=SNPUW(NRDJ)-SMULU*SADWW(N,J)/SNPWW(N)
      SNPWU(NRDJ)=SNPWU(NRDJ)-SMULW*SADWU(N,J)/SNPWW(N)
      SNPWW(NRDJ)=SNPWW(NRDJ)-SMULW*SADWW(N,J)/SNPWW(N)
      DO 35 K=1,JDUM
      IF(NPADJ(NRDJ,K).EQ.(N+NPOUT)) GO TO 35
      DO 36 L=1,IDUM
      LL=L
      IF(NPADJ(N,L).EQ.NPADJ(NRDJ,K)) GO TO 37
   36 CONTINUE
      GO TO 35
   37 SADUU(NRDJ,K)=SADUU(NRDJ,K)-SMULU*SADWU(N,LL)/SNPWW(N)
      SADUW(NRDJ,K)=SADUW(NRDJ,K)-SMULU*SADWW(N,LL)/SNPWW(N)
      SADWU(NRDJ,K)=SADWU(NRDJ,K)-SMULW*SADWU(N,LL)/SNPWW(N)
      SADWW(NRDJ,K)=SADWW(NRDJ,K)-SMULW*SADWW(N,LL)/SNPWW(N)
   35 CONTINUE
      FAU(NRDJ)=FAU(NRDJ)-SMULU*FAW(N)/SNPWW(N)
      FAW(NRDJ)=FAW(NRDJ)-SMULW*FAW(N)/SNPWW(N)
   31 CONTINUE
      FAW(N)=FAW(N)/SNPWW(N)
      DO 38 J=1,IDUM
      SADWU(N,J)=SADWU(N,J)/SNPWW(N)
   38 SADWW(N,J)=SADWW(N,J)/SNPWW(N)
    2 DO 40 J=1,IDUM
      NADJ=NPADJ(N,J)
      NRDJ=NADJ-NPOUT
      ISW=0
      JDUM=NADJNP(NRDJ)
      DO 41 K=1,JDUM
      IF(ISW.EQ.1) GO TO 42
      IF(NPADJ(NRDJ,K).NE.(N+NPOUT)) GO TO 41
      ISW=1
   42 IF(K.EQ.JDUM) GO TO 39
      NPADJ(NRDJ,K)=NPADJ(NRDJ,K+1)
      SADUU(NRDJ,K)=SADUU(NRDJ,K+1)
      SADUW(NRDJ,K)=SADUW(NRDJ,K+1)
      SADWU(NRDJ,K)=SADWU(NRDJ,K+1)
      SADWW(NRDJ,K)=SADWW(NRDJ,K+1)
      GO TO 41
   39 NPADJ(NRDJ,K)=0
      SADUU(NRDJ,K)=0.0
      SADUW(NRDJ,K)=0.0
      SADWU(NRDJ,K)=0.0
      SADWW(NRDJ,K)=0.0
   41 CONTINUE
      NADJNP(NRDJ)=NADJNP(NRDJ)-1
   40 CONTINUE
C
      GO TO 3
C
   15 W(N)=FAW(N)/SNPWW(N)
      U(N)=FAU(N)-SNPUW(N)*W(N)
      GO TO 800
    3 CONTINUE
C
   43 WRITE(10) NPOUT,NRMCP,(ITYPE(I),FAU(I),FAW(I),SNPUW(I),
     1 NADJNP(I),(SADUU(I,J),SADUW(I,J),SADWU(I,J),SADWW(I,J),
```

```
     2NPADJ(I,J),J=1,NXADJP),I=1,NRMCP)
      NUMCP=NUMCP+NRMCP
      NPOUT=NUMCP
      NPR=NUMNPB-NRMCP
      GO TO 902
   44 KX=NPR+1
      GO TO 900
   45 NODESR=NUMNP-NUMCP
      IF(NODESR.LE.MXNPB) NUMNPB=NODESR
      IF(NODESR.GT.MXNPB) NUMNPB=MXNPB
      GO TO 904
   46 NRMCP=0
      GO TO 1
C
  900 DO 901 I=KX,MXNPB
      NADJNP(I)=0
      ITYPE(I)=0
      SNPUU(I)=0.0
      SNPUW(I)=0.0
      SNPWU(I)=0.0
      SNPWW(I)=0.0
      FAU(I)=0.0
      FAW(I)=0.0
      U(I)=0.0
      W(I)=0.0
      DO 901 J=1,NXADJP
      NPADJ(I,J)=0
      SADUU(I,J)=0.0
      SADUW(I,J)=0.0
      SADWU(I J)=0.0
  901 SADWW(I,J)=0.0
      GO TO 45
  902 DO 903 L=1,NPR
      K=NRMCP+L
      NADJNP(L)=NADJNP(K)
      ITYPE(L)=ITYPE(K)
      SNPUU(L)=SNPUU(K)
      SNPUW(L)=SNPUW(K)
      SNPWU(L)=SNPWU(K)
      SNPWW(L)=SNPWW(K)
      FAU(L)=FAU(K)
      FAW(L)=FAW(K)
      U(L)=U(K)
      W(L)=W(K)
      DO 903 J=1,NXADJP
      NPADJ(L,J)=NPADJ(K,J)
      SADUU(L,J)=SADUU(K,J)
      SADUW(L,J)=SADUW(K,J)
      SADWU(L,J)=SADWU(K,J)
  903 SADWW(L,J)=SADWW(K,J)
      GO TO 44
  904 CONTINUE
      DO 905 I=KX,NUMNPB
      READ(14) N,NADJNP(I),ITYPE(I),THETA,XMASS,SNPUU(I),SNPUW(I),
     1SNPWW(I),(NPADJ(I,J),SADUU(I,J),SADUW(I,J),SADWU(I,J),SADWW(I,J),
     2J=1,MXADJP),FAU(I),FAW(I),U(I),W(I)
  905 CONTINUE
      DO 913 I=KX,NUMNPB
      SNPWU(I)=SNPUW(I)
  913 CONTINUE
```

```
      GO TO 46
  909 WRITE(6,910) NPOUT,NODE,NADJ
      WRITE(6,911)  (NPADJ(NODE,K),K=1,NXADJP)
      WRITE(6,911)  (NPADJ(NRDJ,K),K=1,NXADJP)
  910 FORMAT(7H NPOUT=,I5/7H NODE =,I5/7H NADJ =,I5/7H NPADJ=)
  911 FORMAT(16I5)
      CALL EXIT
  800 DO 801 N=1,NJMNP
      UN(N)=0.0
  801 WN(N)=0.0
      UN(NUMNP)=U(NN)
      WN(NUMNP)=W(NN)
C
      IF(NPOUT.EQ.0) GO TO 807
      BACKSPACE 10
  807 CONTINUE
  805 DO 802 I=1,NRMCP
      J=NRMCP+1-I
      N=J+NPOUT
      WN(N)=FAW(J)
      IDUM=NADJNP(J)
      DO 803 K=1,IDUM
      NADJ=NPADJ(J,K)
  803 WN(N)=WN(N)-SADWJ(J,K)*UN(NADJ)-SADWW(J,K)*WN(NADJ)
      UN(N)=FAU(J)-SNPUW(J)*WN(N)
      DO 804 K=1,IDUM
      NADJ=NPADJ(J,K)
  804 UN(N)=UN(N)-SADJJ(J,K)*UN(NADJ)-SADUW(J,K)*WN(NADJ)
C
  802 CONTINUE
C
      IF(N.EQ.1) GO TO 806
C
      READ(10)   NPOUT,NRMCP,(ITYPE(I),FAU(I),FAW(I),SNPUW(I),
     1 NADJNP(I),(SADJJ(I,J),SADUW(I,J),SADWU(I,J),SADWW(I,J),
     2 NPADJ(I,J),J=1,NXADJP),I=1,NRMCP)
      IF(NPOUT.EQ.0) GO TO 805
      BACKSPACE 10
      BACKSPACE 10
      GO TO 805
C
  806 REWIND 10
      REWIND 14
C
      DO 808 I=1,NUMNP
C
      READ(14) N,NTDJNP,JTYPE,THETA,XMASS,STNPUU,STNPUW,STNPWW,
     1 (NPADJ(1,J),SADUU(1,J),SADUW(1,J),SADWU(1,J),SADWW(1,J),
     2 J=1,MXADJP),FU,FW,UDUM,WDUM
C
  808 WRITE(10) N,NTDJNP,JTYPE,THETA,XMASS,STNPUU,STNPUW,STNPWW,FU,FW,
     1(NPADJ(1,J),SADUU(1,J),SADUW(1,J),SADWW(1,J),J=1,MXADJP)
C
      REWIND 8
      REWIND 10
      REWIND 14
C
      DO 809 I=1,NUMNP
  809 READ(8)   J,FAJ(1),FAU(1),(FAU(K),K=1,MXADJP)
      READ(8)   NU,(ITYPE(I),ITYPE(I),ITYPE(I),ITYPE(I),ITYPE(I),
```

```
    1 ITYPE(I),I=1,NJ)
      READ(8)  (NJ,NU,I=1,NUMNP)
      WRITE(8)(UN(I),WN(I),I=1,NUMNP)
C
      REWIND 8
      IF((IPRINT.NE.6).AND.(IPRINT.NE.99))RETURN
      WRITE(6,810)
  810 FORMAT(1H1,22HRESULTS OF ELIMINATION//
     15X,3HNEW,12X,6HJDISPL,14X,6HWDISPL/
     25X,4HNODE,I2X,4H(IN),16X,4H(IN)//)
      DO 811 I=1,NUMNP
  811 WRITE(6,812) I,JN(I),WN(I)
  812 FORMAT(I8,2X,1P2E20.5)
      RETURN
      END
C
C
C
      OVERLAY(MOHAN,11,0     )
      PROGRAM LNK11
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1        NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2        KTAPE,KRJN,IPRINT,NUMST,MXSTRT,IELAST(20),IPLAST(20),
     3 WGT(20),NSTART(79),EI(5,20),IPELTP,INT,NPRCDS,IMPBX
C
      DIMENSION NADJNP(400),ITYPE(400),THETA(400),XMASS(400),
     1SNPUU(400),SNPUW(400),SNPWW(400),NPADJ(400,8),SADUU(400,8),
     2SADUW(400,8),SADWW(400,8),NADJEL(400)
C
C
      DIMENSION STNPU(4,400),STNPW(4,400),STADU(4,400,  ,
     1STADW(4,400,8)
      DIMENSION NPLOW(80),NPHIGH(80),NPOUT(80),NUMCP(80),
     1NELCLS(80),NMPCLS(80),NPTN(1600),FAU(400),FAW(400)
C
      EQUIVALENCE (STNPU(1),ITYPE),(STNPU(401),THETA),
     1(STNPU(801),XMASS),(STNPU(1201),SNPUU),(STNPW(1),SNPUW),
     2(STNPW(401),SNPWW),(STADU(1),SADUU),(STADU(3201),SADUW),
     3(STADU(6401),SADWW),(STADW(1),NPTN),(STNPW(801),FAU),
     4(STNPW(1201),FAW)
C
      MOHAN=5HMOHAN
      REWIND 8
      DO 1 I=1,NUMNP
    1 READ(8) N,NPLOW(1),NPLOW(1),(NPLOW(J),J=1,MXADJP)
      READ(8) NUMCLS,(NPLOW(I),NPHIGH(I),NPOUT(I),NUMCP(I),
     1NELCLS(I),NMPCLS(I),I=1,NUMCLS)
      REWIND 10
      REWIND 14
      IC=1
      NLOW=NPOUT(IC)+1
      NHGH=NUMCP(IC)
  104 DO 101 L=NLOW,NHGH
  101 READ(10) N,NADJNP(L),ITYPE(L),THETA(L),XMASS(L),SNPUU(L),
     1SNPUW(L),SNPWW(L),FAU(L),FAW(L),(NPADJ(L,J),SADUU(L,J),SADUW(L,J),
     2SADWW(L,J),J=1,MXADJP)
      NUMNPB=NUMCP(IC)-NPOUT(IC)
      WRITE(14) NPLOW(IC),NPHIGH(IC),NPOUT(IC),NUMCP(IC),NELCLS(IC),
     1NMPCLS(IC),NUMNPB,(NADJNP(I),ITYPE(I),THETA(I),XMASS(I),SNPUU(I),
     2SNPUW(I),SNPWW(I),FAU(I),FAW(I),(NPADJ(I,J),SADUU(I,J),SADUW(I,J),
```

```
     3SADWW(I,J),J=1,MXADJP),I=1,NUMNPB)
      IC=IC+1
      IF(IC.GT.NUMCLS) GO TO 105
      IF(NPOUT(IC).NE.NUMCP(IC-1)) GO TO 103
      NLOW=1
      NHGH=NUMCP(IC)-NPOUT(IC)
      GO TO 104
  103 NPR=NUMCP(IC-1)-NPOUT(IC)
      DO 102 I=1,NPR
      L=NPOUT(IC)-NPOUT(IC-1)+I
      NADJNP(I)=NADJNP(L)
      ITYPE(I)=ITYPE(L)
      THETA(I)=THETA(L)
      XMASS(I)=XMASS(L)
      SNPUU(I)=SNPJJ(L)
      SNPUW(I)=SNPJW(L)
      SNPWW(I)=SNPWW(L)
      FAU(I)=FAU(L)
      FAW(I)=FAW(L)
      DO 102 J=1,MXADJP
      NPADJ(I,J)=NPADJ(L,J)
      SADUU(I,J)=SADUJ(L,J)
      SADUW(I,J)=SADUW(L,J)
  102 SADWW(I,J)=SADWW(L,J)
      NLOW=NUMCP(IC-1)-NPOUT(IC)+1
      NHGH=NUMCP(IC)-NPOUT(IC)
      GO TO 104
  105 REWIND 10
      REWIND 14
      DO 106 NC=1,NUMCLS
      READ(14) N1,N2,N3,N4,N5,N6,N7,(NADJNP(I),ITYPE(I),THETA(I),
     1XMASS(I),SNPJJ(I),SNPUW(I),SNPWW(I),FAU(I),FAW(I),(NPADJ(I,J),
     2SADUU(I,J),SADJW(I,J),SADWW(I,J),J=1,MXADJP),I=1,N7)
      WRITE(10)N1,N2,N3,N4,N5,N6,N7,(NADJNP(I),ITYPE(I),THETA(I),
     1XMASS(I),SNPJJ(I),SNPUW(I),SNPWW(I),FAU(I),FAW(I),(NPADJ(I,J),
     2SADUU(I,J),SADJW(I,J),SADWW(I,J),J=1,MXADJP),I=1,N7)
  106 CONTINUE
      READ(8)     (NPTN(I),NC,I=1,NUMNP)
      WRITE(10)   (NPTN(I),I=1,NUMNP)
      REWIND 8
      REWIND 3
      IC=1
      NLOW=1
      NHGH=NUMCP(IC)
  204 DO 201 L=NLOW,NHGH
  201 READ(3) N,NADJNP(L),NADJEL(L),(NPADJ(L,J),J=1,MXADJP),
     1(STNPU(K,L),STNPW(K,L),K=1,4),((STADU(K,L,J),STADW(K,L,J),
     2K=1,4),J=1,MXADJP)
C
      NUMNPB=NUMCP(IC)-NPOUT(IC)
      WRITE(10) NPLOW(IC),NPHIGH(IC),NPOUT(IC),NUMCP(IC),NELCLS(IC),
     1NMPCLS(IC),NUMNPB,(NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP),
     2(STNPJ(K,I),STNPW(K,I),K=1,4),((STADU(K,I,J),STADW(K,I,J),
     3K=1,4),J=1,MXADJP),I=1,NUMNPB)
C
      IC=IC+1
      IF(IC.GT.NUMCLS) GO TO 205
      IF(NPOUT(IC).NE.NUMCP(IC-1)) GO TO 203
      NLOW=1
      NHGH=NUMCP(IC)-NPOUT(IC)
```

```
      GO TO 204
C
  203 NPR=NUMCP(IC-1)-NPOUT(IC)
      DO 202 I=1,NPR
      L=NPOUT(IC)-NPOUT(IC-1)+I
      NADJNP(I)=NADJNP(L)
      NADJEL(I)=NADJEL(L)
      DO 207 J=1,MXADJP
  207 NPADJ(I,J)=NPADJ(L,J)
      DO 202 K=1,4
      STNPU(K,I)=STNPU(K,L)
      STNPW(K,I)=STNPW(K,L)
      DO 202 J=1,MXADJP
      STADU(K,I,J)=STADU(K,L,J)
  202 STADW(K,I,J)=STADW(K,L,J)
C
      NLOW=NUMCP(IC-1)-NPOUT(IC)+1
      NHGH=NUMCP(IC)-NPOUT(IC)
      GO TO 204
C
  205 REWIND 10
      REWIND 3
      RETURN
      END
C
C
C
      OVERLAY(MOHAN,12,0     )
      PROGRAM LNK1J
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1       NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2       KTAPE,KRJN,IPRINT,NUMST,MXSTRT,IELAST(20),IPLAST(20),
     3 WGT(20),NSTART(79),EI(5,20),IPELTP,INT,NPRCDS,IMPBX
      DIMENSION NPLOW(80),NPHIGH(80),NPOUT(80),NUMCP(80),
     1NELCLS(80),NMPCLS(80)
C
      DIMENSION JPLAST(20),CC(4,4)
      DIMENSION NOOFEL(24),NP(24,4),ITYPE(24,4),THETA(24,4),
     1C(24,4,4),B(24,4,8),P(24,8,4),EPSTI1(24,4),EPSPI1(24,4),
     2SIGI1(24,4),DUM(24,29),IDUM(24,29)
      EQUIVALENCE (DUM,IDUM)
      DIMENSION ALPHA(20),CAPPA(20),COSTH(20),NOYILD(20),
     1           SSTAR(20,10),HSTAR(20,10)
     2    ,COHESN(20),FRCTN1(20),FRCTN2(20),SNSWCH(20),
     3    CRESID(20),FRESID(20),JTENSN(20),
     4    MYIELD(20),TRESID(20)
      MOHAN=5HMOHAN
      IF(NUMPEL.EQ.0) RETURN
      REWIND 12
      REWIND 8
      DO 1 I=1,NUMNP
    1 READ(8) I,NPLOW(I),NPLOW(I),(NPLOW(J),J=1,MXADJP)
      READ(8) NUMCLS,(NPLOW(I),NPHIGH(I),NPOUT(I),NUMCP(I),
     1NELCLS(I),NMPCLS(I),I=1,NUMCLS)
      REWIND 8
      REWIND 4
      DO 2 I=1,NUMNP
    2 READ(4) N,R,D,TT,TH
      READ(4) NZONES
      DO 3 I=1,NZONES
```

```
      READ(4) IZ,IELAST(IZ),JPLAST(IZ),WGT(IZ),(EI(J,IZ),J=1,5)
      IF(JPLAST(IZ).EQ.0) GO TO 3
      IF(JPLAST(IZ).GT.1) GO TO 4
      READ(4) K,(SSTAR(IZ,J),J=1,K),(HSTAR(IZ,J),J=1,K)
      NOYILD(IZ)=K
      GO TO 3
    4 IF(JPLAST(IZ).GT.2) GO TO 10
      READ( 4) ALPHA(IZ),CAPPA(IZ),COSTH(IZ)
      GO TO 3
   10 IF(JPLAST(IZ).GT.3) GO TO 5
      READ(4)COHESN(IZ),FRCTN1(IZ),FRCTN2(IZ),
     1        SNSWCH(IZ),CRESID(IZ),FRESID(IZ),
     2        MYIELD(IZ),IRESID(IZ),JTENSN(IZ)
      GO TO 3
    5 WRITE(6,6) IZ,JPLAST(IZ)
    6 FORMAT(1H1,22HERROR IN L1F ZONE DATA//10X,7HZONE   =,I5/
     110X,7HJPLAST=,I5)
      CALL EXIT
    3 CONTINUE
C
      JCLUS=1
      NUMTEL=0
    8 IF(NMPCLS(JCLUS).NE.0) GO TO 7
      JCLUS=JCLUS+1
    9 IF(JCLUS.LE.NJMCLS) GO TO 8
      REWIND 12
      REWIND 4
      RETURN
C
    7 NUMCEL=0
  100 IF((NELCLS(JCLUS)-NUMCEL).LT.MXPELB) NUMELB=NELCLS(JCLUS)-NUMCEL
      IF((NELCLS(JCLUS)-NUMCEL).GE.MXPELB) NUMELB=MXPELB
      DO 200 KK=1,NJMELB
      READ(4) JJ,NOOFEL(KK),IZONE,IPLAST(KK),NP(KK,1),NP(KK,2),NP(KK,3),
     1NP(KK,4),NCRACK,ITYPE(KK,1),ITYPE(KK,2),ITYPE(KK,3),ITYPE(KK,4),
     2          THETA(KK,1),THETA(KK,2),THETA(KK,3),THETA(KK,4),
     3          RI,RJ,RK,RL,ZI,ZJ,ZK,ZL
      IF(JJ.EQ.JCLUS) GO TO 239
      WRITE(6,240) JJ,JCLUS
  240 FORMAT(1H1,7HJJ     =,I5/7H JCLUS=,I5)
      CALL EXIT
  239 CONTINUE
  202 IE=IELAST(IZONE)
      A1=EI(1,IZONE)
      A2=EI(2,IZONE)
      A3=EI(3,IZONE)
      A4=EI(4,IZONE)
      A5=EI(5,IZONE)
      NUME=NOOFEL(KK)
      CALL ELCST(IE,ISTRES,A1,A2,A3,A4,A5,CC,NUME)
      DO 203 I=1,4
      DO 203 J=1,4
  203 C(KK,I,J)=CC(I,J)
C
      DO 204 I=1,4
      DO 204 J=1,8
      B(KK,I,J)=0.0
  204 P(KK,J,I)=0.0
C
      IF(NP(KK,4).NE.0) GO TO 208
```

```
C
      AJ=RJ-RI
      AK=RK-RI
      BJ=ZJ-ZI
      BK=ZK-ZI
      HH=AJ*BK-AK*BJ
      AA=AJ-AK
      BB=BJ-BK
C
      B(KK,1,1)=BB/HH
      B(KK,1,3)=BK/HH
      B(KK,1,5)=-BJ/HH
      IF(ISTRES.NE.0) GO TO 205
      RO=(AJ+AK)/3.
      ZO=(BJ+BK)/3.
      CAPRO=RI+RO
      B(KK,2,1)=(HH+BB*RO-AA*ZO)/(HH*CAPRO)
      B(KK,2,3)=(BK*RO-AK*ZO)/(HH*CAPRO)
      B(KK,2,5)=(-BJ*RO+AJ*ZO)/(HH*CAPRO)
  205 B(KK,3,2)=-AA/HH
      B(KK,3,4)=-AK/HH
      B(KK,3,6)=AJ/HH
      B(KK,4,1)=B(KK,3,2)
      B(KK,4,2)=B(KK,1,1)
      B(KK,4,3)=B(KK,3,4)
       B(KK,4,4)=B(KK,1,3)
      B(KK,4,5)=B(KK,3,6)
      B(KK,4,6)=B(KK,1,5)
C
      IF(ISTRES.NE.0) CONST=HH/2.
      IF(ISTRES.EQ.0) CONST=HH*CAPRO/2.
  212 DO 206 I=1,8
      DO 206 J=1,4
      DO 207 N=1,4
  207 P(KK,I,J)=P(KK,I,J)+CONST*B(KK,N,I)*CC(N,J)
  206 CONTINUE
      GO TO 231
C
  208 AJ=RJ-RI
      BJ=ZJ-ZI
      AA=SQRT(AJ*AJ+BJ*BJ)
      AL=RL-RI
      BL=ZL-ZI
      BB=SQRT(AL*AL+BL*BL)
      HH=AA*BB
      S1=-BJ/AA
      C1=AJ/AA
      IF(NCRACK.EQ.0) GO TO 213
      HH=AA
      BB=0.0
  213 CONTINUE
      IF(ISTRES.EQ.0) GO TO 209
      RO=AA/2.
      ZO=BB/2.
      GO TO 210
  209 AINT1=HH
      AINT2=HH*BB/2.
      AINT3=HH*AA/2.
      AINT4=HH*(RI+(AA*C1+BB*S1)/2.)
      AINT13=AA*AINT4/2.+HH*AA**2*C1/12.
```

```
      AINT14=BB*AINT4/2.+HH*BB**2*S1/12.
      AINT16=(HH/2.)**2
      RC=AINT13/AINT4
      ZC=AINT14/AINT4
  210 DUMMY=RC*S1+ZO*C1
      B(KK,1,1)=(-AA*S1-BB*C1 +DUMMY)/HH
      B(KK,1,3)=(BB*C1-DUMMY)/HH
      B(KK,1,5)=DUMMY/HH
      B(KK,1,7)=(AA*S1-DUMMY)/HH
      IF(ISTRES.NE.0) GO TO 211
      IF(NCRACK.EQ.1) GO TO 211
      B(KK,2,1)=(HH*AINT1-BB*AINT3+AINT16-AA*AINT2)/(HH*AINT4)
      B(KK,2,3)=(BB*AINT3-AINT16)/(HH*AINT4)
      B(KK,2,5)=AINT16/(HH*AINT4)
      B(KK,2,7)=(AA*AINT2-AINT16)/(HH*AINT4)
  211 DUMMY=RC*C1-ZO*S1
      B(KK,3,2)=((BB*S1-AA*C1)+DUMMY)/HH
      B(KK,3,4)=(-BB*S1-DUMMY)/HH
      B(KK,3,6)=DUMMY /HH
      B(KK,3,6)=DUMMY /HH
      B(KK,3,8)=(AA*C1-DUMMY)/HH
      B(KK,4,1)=B(KK,3,2)
      B(KK,4,2)=B(KK,1,1)
      B(KK,4,3)=B(KK,3,4)
      B(KK,4,4)=B(KK,1,3)
      B(KK,4,5)=B(KK,3,6)
      B(KK,4,6)=B(KK,1,5)
      B(KK,4,7)=B(KK,3,8)
      B(KK,4,8)=B(KK,1,7)
      IF(ISTRES.NE.0) CONST=H'
      IF(ISTRES.EQ.0) CONST=AINT4
      GO TO 212
C
  231 DO 232 I=1,4
      EPSTI1(KK,I)=0.0
      EPSPI1(KK,I)=0.0
  232 SIGI1(KK,I)=0.0
      IF(IPLAST(KK).NE.1) GO TO 237
      IDUM(KK,21)=NOYILD(IZONE)
      KYILD=NCYILD(IZONE)
      DO 233 I=1,KYILD
      DUM(KK,I)=SSTAR(IZONE,I)
  233 DUM(KK,I+10)=HSTAR(IZONE,I)
      DO 234 I=1,8
  234 DUM(KK,I+21)=0.0
      DUM(KK,27)=SSTAR(IZONE,1)
      GO TO 200
  237 IF(IPLAST(KK).NE.2) GO TO 241
C
      DUM(KK,1)=ALPHA(IZONE)
      DUM(KK,2)=CAPPA(IZONE)
      DUM(KK,3)=COSTH(IZONE)
      IDUM(KK,4)=0
      IF(CAPPA(IZONE).EQ.0.0) IDUM(KK,4)=1
      DUM(KK,5)=0.0
      DO 238 I=1,4
  238 DUM(KK,I+5)=0.0
      GO TO 200
  241 IF(IPLAST(KK).NE.3) GO TO 235
      DUM(KK,1) = COHESN(IZONE)
```

```
      DUM(KK,2) = FRCTN1(IZONE)
      DUM(KK,3) = FRCTN2(IZONE)
      DUM(KK,4) = SNSWCH(IZONE)
      DUM(KK,5) = CRESID(IZONE)
      DUM(KK,6) = FRESID(IZONE)
      IDUM(KK,7) = MYIELD(IZONE)
      IDUM(KK,8) = IRESID(IZONE)
      IDUM(KK,9) = JTENSN(IZONE)
      AJ=RJ-RI
      BJ=ZI-ZJ
      AL=SQRT(AJ*AJ+BJ*BJ)
      CO=AJ/AL
      SI=BJ/AL
      DUM(KK,10) = CO
      DUM(KK,11) = SI
      GO TO 200
C
  235 WRITE(6,236) NOOFEL(K),IZONE,IPLAST(KK)
  236 FORMAT(1H1,28HERROR IN ELEMENT DATA, L1H  //
     110X,12HELEMENT NO.=,I5/10X,12HZONE NO.    =,I5/
     210X,12HIPLAST      =,I5)
      CALL EXIT
C
  200 CONTINUE
C
      NUMCEL=NUMCEL+NUMELB
      NUMTEL=NUMTEL+NUMELB
      EFFECT=0.0
      WRITE(12) NUMTEL,NUMELB,(NOOFEL(K),IPLAST(K),(NP(K,J),ITYPE(K,J),
     1THETA(K,J),J=1,4),((C(K,J,I),I=1,4),(B(K,J,I),I=1,8),J=1,4),
     2((P(K,J,I),J=1,8),EPSTI1(K,I),EPSPI1(K,I),SIGI1(K,I),I=1,4),
     3(DUM(K,I),I=1,29),EFFECT,K=1,NUMELB)
      IF(NUMCEL.LT.NELCLS(JCLUS)) GO TO 100
      JCLUS=JCLUS+1
      GO TO 9
      END
C
C
C
      SUBROUTINE ELOSTI(IELAST,ISTRES,E1,E2,E3,E4,E5,C,NUME)
      DIMENSION C(4,4)
C**** FORM STRESS-STRAIN MATRIX
      DO 1 I=1,4
      DO 1 J=1,4
    1 C(I,J)=0.0
      IF(IELAST.NE.1) GO TO 20
C**** ISOTROPIC ELASTIC MATERIAL
      IF(ISTRES.EQ.2) GO TO 4
C     AXISYMMETRIC OR PLANE STRAIN PROBLEM
      EBAR=E1/((1.+E2)*(1.-2.*E2))
      C(1,1)=EBAR*(1.-E2)
      C(1,2)=EBAR*E2
      C(1,3)=C(1,2)
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=C(1,2)
      C(3,1)=C(1,2)
      C(3,2)=C(1,2)
      C(3,3)=C(1,1)
      C(4,4)=EBAR*(1.-2.*E2)/2.
```

```
      RETURN
C     PLANE STRESS PROBLEM
C
    4 EBAR=E1/(1.-E2*E2)
      C(1,1)=EBAR
      C(3,1)=EBAR*E2
      C(1,3)=C(3,1)
      C(3,3)=C(1,1)
      C(4,4)=EBAR*(1.-E2)/2.
      RETURN
C**** ANISOTROPIC ELASTIC MATERIAL
C
   20 IF(IELAST.NE.2) GO TO 30
      IF(ISTRES.EQ.2) GO TO 2
      C(1,1)=E1
      C(1,2)=E1-2.*E5
      C(1,3)=E3
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=C(1,3)
      C(3,1)=C(1,3)
      C(3,2)=C(2,3)
      C(3,3)=E2
      C(4,4)=E4
      RETURN
C
    2 C(1,1)=2.*E5*(E1-2.*E5)/E1
      C(1,3)=2.*E3*E5/E1
      C(3,1)=C(1,3)
      C(3,3)=E2-E3**2/E1
      C(4,4)=E4
      RETURN
   21 WRITE(6,3) IELAST,NUME,ISTRES
    3 FORMAT(1H1/31H ERROR IN ELASTIC CONSTANT DATA/
     113H IELAST     =,I5/13H ELEMENT NO.=,I5/
     213H ISTRES     =,I5)
      CALL EXIT
C
   30 IF(IELAST.NE.3) GO TO 21
C**** COMPRESSIBLE FLUID
      IF(ISTRES.EQ.2) GO TO 21
      DO 31 I=1,3
      DO 31 J=1,3
   31 C(I,J)=E1
      RETURN
C
      END
C
C
C
      OVERLAY(MOHAN,13,0     )
      PROGRAM LNK2
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPEL3,NUMNP,
     1       NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2      KTAPE,KRUN,IPRINT,NUMST,MXSTRT,FUZ(239),
     3                          IPELTP,INT,NPRCDS,IMPBX
      COMMON/A/ U(1600),W(1600),NPOUT(80),NMPCLS(80),FNU(350),
     1       FNW(350)
      COMMON/B/ NADJNP(400),ITYPE(400),THETA(400),XMASS(400),
     1    SNPUU(400),SNPUW(400),SNPWW(400),FAU(400),FAW(400),
```

```
     2   NPADJ(400,6),SADUU(400,8),SADUW(400,8),SADWW(400,8)
      DIMENSION ANAME(18),COM(1)
C
      EQUIVALENCE (MAXNP,COM(1))
      MOHAN=5HMOHAN
      IPELTP=12
      INT=8
      NPRCDS=0
      REWIND INT
      REWIND IPELTP
C
      READ(5,1) ITMAX,ERRMAX,NFAC,KTAPE,ICONTU,OVERLX
    1 FORMAT(I5,E10.0,3I5,E10.0)
C     ITMAX =MAX. NO. OF ITERATIONS PER SOLUTION
C     ERRMAX=MAX. ALLOWABLE ERROR (LBS)
C     NFAC  =INCREMENTS FOR NONLINEAR SOLUTION
C     KTAPE =0 USE TWO K TAPES
C            =1 USE ONE K TAPE
C     ICONTU=0 CONTINUE SOLUTION FOR NONCONVERGENCE
C     OVERLX=OVER-RELAXATION FACTOR
      FACTOR=FLOAT(NFAC)
      WRITE(6,2) ITMAX,ERRMAX,FACTOR,KTAPE,ICONTU,OVERLX
    2 FORMAT(1H1,7HITMAX =,I5/8H ERRMAX=,1PE15.5/8H FACTOR=,1PE15.5/
     18H KTAPE =,I5/8H ICONTU=,I5/8H OVERLX=,1PE15.5//)
      IF(OVERLX.LE.0.0) CALL EXIT
      REWIND 8
      DO 3 I=1,NUMNP
    3 READ(8) N,NPOUT(1),NPOUT(1),(NPOUT(J),J=1,MXADJP)
      READ(8) NMKCLS,(NPLOW,NPHIGH,NPOUT(I),NUMCP,NELCLS,NMPCLS(I),
     1I=1,NMKCLS)
      READ(8) (NPLOW,NPLOW,I=1,NUMNP)
      READ(8) (U(I),W(I),I=1,NUMNP)
      REWIND 8
   16 WRITE(6,17)
   17 FORMAT(1H1,16HELASTIC SOLUTION//4X,6HKERROR,4X,6HNERROR,14X,
     16HERRCNT,14X,6HERRMAX,4X,6HICOUNT,5X,5HITMAX,7X,3HINT,4X,
     26HIPELTP//)
      ICOUNT=1
   18 CALL ERROR(KERROR,ERRMAX,1,NERROR,ERRCNT,OVERLX)
      WRITE(6,19) KERROR,NERROR,ERRCNT,ERRMAX,ICOUNT,ITMAX,INT,IPELTP
   19 FORMAT(2I10,1P2E20.5,4I10)
      ALAMB=0.0
      IF(KERROR.EQ.0) GO TO 22
      IF(ICOUNT.EQ.ITMAX) GO TO 20
      IF(ICOUNT.EQ.ITMAX) GO TO 20
      ICOUNT=ICOUNT+1
      GO TO 18
   20 WRITE(6,21)
   21 FORMAT(//39H HAVE NOT CONVERGED TO ELASTIC SOLUTION//)
   22 IF(NUMPEL.EQ.0) GO TO 27
      DO 23 ICLUS=1,NMKCLS
      IF(NMPCLS(ICLUS).EQ.0) GO TO 23
      NUM=NMPCLS(ICLUS)
      DO 24 I=1,NUM
   24 CALL PLASTF(0,0,ICLUS,0,I)
   23 CONTINUE
      WRITE(6,25) ALAMB
   25 FORMAT(//21H ELASTIC LOAD FACTOR=,1PE20.5//)
      IF(ALAMB.LE.1.0) GO TO 27
      IF(ALAMB.LT.1.0E+38) GO TO 54
```

```
      AA=0.0
      WRITE(INT) (COM(I), I=1, 16)
      DO 55 I=1,NJMNP
   55 WRITE(INT) AA,AA
      ALAMB=2.0
      GO TO 56
   54 CONTINUE
      DO 26 I=1,NJMNP
      U(I)=U(I)/ALAMB
   26 W(I)=W(I)/ALAMB
C
   27 WRITE(INT) (COM(I), I=1, 16)
      DO 38 I=1,NJMNP
   38 WRITE(INT) U(I),W(I)
   53 IF(NUMPEL.EQ.0) RETURN
      DO 42 ICLUS=1,NMKCLS
      IF(NMPCLS(ICLUS).EQ.0) GO TO 42
      NUM=NMPCLS(ICLUS)
      DO 43 I=1,NUM
   43 CALL PLASTF(1,1,ICLUS,0,I)
   42 CONTINUE
      IF(ALAMB.LE.1.0) RETURN
   56 IF(KERRCR.EQ.0) GO TO 52
      IF(ICONTU.NE.0) RETURN
   52 CONTINUE
      FAC=(ALAMB-1.0)/FACTOR
      DO 28 I=1,NJMNP
      U(I)=U(I)*FAC
   28 W(I)=W(I)*FAC
C
      DO 36 IFAC=1,NFAC
      ICOUNT=1
   32 CALL ERROR(KERROR,ERRMAX,C,NERROR,ERRCNT,OVERLX)
      WRITE(6,19) KERROR,NERROR,ERRCNT,ERRMAX,ICOUNT,ITMAX,INT,IPELTP
C
      IF(KERRCR.NE.0) GO TO 31
   35 DO 41 I=1,NJMNP
   41 WRITE(INT) U(I),W(I)
      DO 29 ICLUS=1,NMKCLS
      IF(NMPCLS(ICLUS).EQ.0) GO TO 29
      NUM=NMPCLS(ICLUS)
      DO 30 I=1,NUM
   30 CALL PLASTF(1,1,ICLUS,IFAC,I)
   29 CONTINUE
      WRITE(6,37) IFAC
   37 FORMAT(//32H HAVE FINISHED PLASTIC INCREMENT,I5//)
      NPRCDS=NPRCDS+1
      IF(KERRCR.EQ.0) GO TO 36
      IF(ICONTU.NE.0) RETURN
      GO TO 36
   31 IF(ICOUNT.EQ.ITMAX) GO TO 33
      ICOUNT=ICOUNT+1
      GO TO 32
   33 WRITE(6,34) IFAC
   34 FORMAT(//57H HAVE NOT CONVERGED TO PLASTIC SOLUTION FOR INCREMENT
     1NC.,I5//)
      GO TO 35
   36 CONTINUE
C
      REWIND IPELTP
```

```
      REWIND INT
      RETURN
      END
C
C
C
      SUBROUTINE ERROR(KERROR,ERRMAX,KSWTCH,NERROR,ERRCNT,OVERLX)
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1       NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2      KTAPE,KRJN,IPRINT,NUMST,MXSTRT,FUZ(239),
     3                        IPELTP,INT,NPRCDS,IMPBX
      COMMON/A/ U(1600),W(1600),NPOUT(80),NMPCLS(80),FNU(350),
     1     FNW(350)
      COMMON/B/NADJNP(400),ITYPE(400),THETA(400),XMASS(400),
     1   SNPUU(400),SNPUW(400),SNPWW(400),FAU(400),FAW(400),
     2   NPADJ(400,8),SADUU(400,8),SADUW(400,8),SADWW(400,8)
      DATA IRT/1/
C     KSWTCH=0 DC NONLINEAR PART OF ANALYSIS
C           =1 DC ELASTIC ANALYSIS ONLY
      IF(IRT.EQ.0) GO TO 100
      IRT=0
      ISWTCH=0
      JSWTCH=0
      REWIND 10
      REWIND 1
      DO 5 I=1,MXNPB
      FNU(I)=0.0
    5 FNW(I)=0.0
      IF((NUMNP.LE.MXNPB).AND.(NUMPEL.EQ.0)) ISWTCH=1
      IF(KTAPE.NE.0) JSWTCH=1
      IC=0
    1 IC=IC+1
      READ(10)       NPLOW,NPHIGH,NPOUT(IC),NUMCP,NELCLS,NMPCLS(IC),
     1NUMNPB,(NADJNP(I),ITYPE(I),THETA(I),XMASS(I),SNPUU(I),SNPUW(I),
     2SNPWW(I),FAU(I),FAW(I),(NPADJ(I,J),SADUU(I,J),SADUW(I,J),
     3SADWW(I,J),J=1,MXADJP),I=1,NUMNPB)
      IF(ISWTCH.EQ.1) GO TO 4
      IF(JSWTCH.EQ.1) GO TO 2
      WRITE(1)       NPLOW,NPHIGH,NPOUT(IC),NUMCP,NELCLS,NMPCLS(IC),
     1NUMNPB,(NADJNP(I),ITYPE(I),THETA(I),XMASS(I),SNPUU(I),SNPUW(I),
     2SNPWW(I),FAU(I),FAW(I),(NPADJ(I,J),SADUU(I,J),SADUW(I,J),
     3SADWW(I,J),J=1,MXADJP),I=1,NUMNPB)
    2 IF(NUMCP.LT.NUMNP) GO TO 1
      REWIND 10
      REWIND 1
      IF(JSWTCH.EQ.1) GO TO 3
      IO=1
      GO TO 100
    3 IO=10
    4 REWIND 10
      IC=10
  100 IF((ISWTCH.EQ.1).OR.(JSWTCH.EQ.1)) GO TO 101
      IF(IO.EQ.10) GO TO 102
      IO=10
      GO TO 103
  102 IO=1
  103 CONTINUE
  101 KERROR=0
      NERROR=0
      ERRCNT=0.0
```

```
      DC 104 ICLUS=1,NMKCLS
      IF(KSWTCH.EQ.1) GO TO 113
      IF(NUMPEL.EQ.0) GO TO 113
      IF(NMPCLS(ICLUS).EQ.0) GO TO 113
      DO 114 I=1,MXNPB
      FNU(I)=0.0
  114 FNW(I)=0.0
      NUM=NMPCLS(ICLUS)
      DO 115 I=1,NUM
  115 CALL PLASTF(0,1,ICLUS,0,I)
  113 IF(ISWTCH.EQ.1) GO TO 105
      READ(IO)      NPLOW,NPHIGH,NPO      ,NUMCP,NELCLS,NMP         ,
     1NUMNPB,(NADJNP(I),ITYPE(I),THETA(I),XMASS(I),SNPUU(I),SNPUW(I),
     2SNPWW(I),FAU(I),FAW(I),(NPADJ(I,J),SADUU(I,J),SADUW(I,J),
     3SADWW(I,J),J=1,MXADJP),I=1,NUMNPB)
  105 NLOW=NPLOW-NPOUT(ICLUS)
      NHGH=NPHIGH-NPOUT(ICLUS)
      IF(KSWTCH.EQ.0) FAC=(1.0-1.0/ALAMB)/FACTOR
      IF(KSWTCH.EQ.1) FAC=1.0
      DO 106 I=NLOW,NHGH
      ERRU=0.0
      ERRW=0.0
      L=I+NPOUT(ICLUS)
      IF(ITYPE(I).EQ.2) GO TO 106
      IF(ITYPE(I).EQ.3) GO TO 108
      ERRU=FNU(I)+FAU(I)*FAC
      ERRU=ERRU-SNPUU(I)*U(L)-SNPUW(I)*W(L)
      NUM=NADJNP(I)
      DO 107 J=1,NUM
      NP=NPADJ(I,J)
  107 ERRU=ERRU-SADUU(I,J)*U(NP)-SADUW(I,J)*W(NP)
      IF(ABS(ERRU).GT.ERRCNT) ERRCNT=ABS(ERRU)
      IF(ABS(ERRU).LE.ERRMAX) GO TO 116
      IF(ABS(ERRU).LE.ERRMAX) GO TO 116
      KERROR=1
      NERROR=NERROR+1
 116  U(L)=U(L) + OVERLX*ERRU/SNPUU(I)
      IF(ITYPE(I).EQ.1) GO TO 106
  108 ERRW=FNW(I)+FAW(I)*FAC
      ERRW=ERRW-SNPUW(I)*U(L)-SNPWW(I)*W(L)
      NUM=NADJNP(I)
      DO 109 J=1,NUM
      NP=NPADJ(I,J)
      NPR=NP-NPOUT(ICLUS)
      DC 110 K=1,MXADJP
      KK=K
      IF(NPADJ(NPR,K).EQ.L) GO TO 112
  110 CONTINUE
      WRITE(6,111) ICLUS,NLOW,NHGH,NPOUT(ICLUS),I,L,NP,NPR
  111 FORMAT(1H1,24HERROR IN COMPUTING SADWU//8I5)
      CALL EXIT
  112 SADWU=SADUW(NPR,KK)
  109 ERRW=ERRW-SADWU*U(NP)-SADWW(I,J)*W(NP)
      IF(ABS(ERRW).GT.ERRCNT) ERRCNT=ABS(ERRW)
      IF(ABS(ERRW).LE.ERRMAX) GO TO 117
      KERROR=1
      NERROR=NERROR+1
 117  W(L) = W(L) + OVERLX*ERRW/SNPWW(I)
  106 CONTINUE
  104 CONTINUE
```

```
      REWIND IO
      RETURN
      END
      SUBROUTINE PLASTF(LSWTCH,MSWTCH,ICLUS,IFAC,INUM)
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPEL3,NUMNP,
     1       NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2      KTAPE,KRUN,IPRINT,NUMST,MXSTRT,FUZ(239),
     3                        IPELTP,INT,NPRCDS,IMPBX
      COMMON/A/ U(1600),W(1600),NPOUT(80),NMPCLS(80),FNU(350),
     1       FNW(350)
      COMMON/B/NADJNP(400),JTYPE(400),SHETA(400),XMASS(400),
     1   SNPUU(400),SNPUW(400),SNPWW(400),FAU(400),FAW(400),
     2   NPADJ(400,8),SADUU(400,8),SADUW(400,8),SADWW(400,8)
C
      DIMENSION BUFF(3280)
      EQUIVALENCE (BUFF,NADJNP)
      DIMENSION NOOFEL(24),IPLAST(24),NP(24,4),ITYPE(24,4),THETA(24,4),
     1C(24,4,4),B(24,4,8),P(24,8,4),EPSTI1(24,4),EPSPI1(24,4),
     2SIGI1(24,4),DUM(24,29),IDUM(24,29)
      EQUIVALENCE (BUFF(1),NOOFEL),(BUFF(25),IPLAST),(BUFF(49),NP),
     1(BUFF(145),ITYPE),(BUFF(241),THETA),(BUFF(337),C),(BUFF(721),B),
     2(BUFF(1489),P),(BUFF(2257),EPSTI1),(BUFF(2353),EPSPI1),
     3(BUFF(2449),SIGI1),(BUFF(2545),DUM),(BUFF(2545),IDUM)
      DIMENSION EPSTI(4),EPSPI(4),SIGI(4),EPSDI(4),X(8),FPLAST(8),
     1EFFECT(24)
      DIMENSION STRESS(4),STRAIN(4),PSTRAN(4),CMAT(4,4),FMAT(4,4),
     1 GMAT(4,4),SIGNI1(4),SIGNBI(4)
C     LSWTCH=0 DO NOT UPDATE ELEMENT TAPE
C           =1        UPDATE ELEMENT TAPE
C     MSWTCH=0 FIND ALAMB VALUE ONLY
C           =1 DO ALL NONLINEAR PART
      IO=IPELTP
      IF(IO.EQ.12) JO=3
      IF(IO.EQ. 3) JO=12
      L=I+NPOUT(ICLUS)
      READ(IO)  NUMCEL,NELBUF,(NOOFEL(K),IPLAST(K),(NP(K,J),
     1ITYPE(K,J),THETA(K,J),J=1,4),((C(K,J,I),I=1,4),(B(K,J,I),I=1,8),
     2J=1,4),((P(K,J,I),J=1,8),EPSTI1(K,I),EPSPI1(K,I),SIGI1(K,I),
     3I=1,4),(DUM(K,I), I=1,29),EFFECT(K),K=1,NELBUF)
      DO 3 I=1,NELBUF
      DO 5 J=1,4
      N=NP(I,J)
      K=2*J-1
      IF(J.EQ.4) GO TO 6
    8 IF(ITYPE(I,J).EQ.1) GO TO 7
      X(K)=U(N)
      X(K+1)=W(N)
      GO TO 5
    7 X(K)=U(N)*COS(THETA(I,J))-W(N)*SIN(THETA(I,J))
      X(K+1)=U(N)*SIN(THETA(I,J))+W(N)*COS(THETA(I,J))
      GO TO 5
    6 IF(N.NE.0) GO TO 8
      X(7)=0.0
      X(8)=0.0
    5 CONTINUE
      DO 19 J=1,8
      IF(X(J).EQ.0.0) GO TO 19
      GO TO 18
   19 CONTINUE
      GO TO 3
```

```
   18 DO 9 J=1,4
      EPSTI(J)=EPSTI1(I,J)
      DO 9 K=1,8
    9 EPSTI(J)=EPSTI(J)+B(I,J,K)* X(K)
      IF(IPLAST(I).NE.1) GO TO 200
      L=I
      CALL MISES(L,EPSTI,EPSPI,SIGI,EPSDI,EEFFI,SYI,SMAXI,SI,MSWTCH,
     1BLAMB)
      IF(MSWTCH.EQ.0) GO TO 400
      IF(LSWTCH.EQ.0) GO TO 100
      DO 101 J=1,4
  101 DUM(I,J+21)=EPSDI(J)
      DUM(I,26)=EEFFI
      DUM(I,27)=SYI
      DUM(I,28)=SMAXI
      DUM(I,29)=SI
  100 CONTINUE
      GO TO 300
  200 IF(IPLAST(I).NE.2) GO TO 500
      L=I
      CALL COULMR(L,EPSTI,EPSPI,SIGI,KORNER,FYLDI,EPSDI,MSWTCH,BLAMB)
      IF(MSWTCH.EQ.0) GO TO 400
      IF(LSWTCH.EQ.0) GO TO 201
      IDUM(I,4)=KORNER
      DUM(I,5)=FYLDI
      DO 202 J=1,4
  202 DUM(I,J+5)=EPSDI(J)
  201 CONTINUE
      GO TO 300
  500 IF(IPLAST(I).NE.3) GO TO 900
      L=I
      COHESN=DUM(I,1)
      FRCTN1=DUM(I,2)
      FRCTN2=DUM(I,3)
      SNSWCH=DUM(I,4)
      CRESID=DUM(I,5)
      FRESID=DUM(I,6)
      MYIELD=IDUM(I,7)
      IRESID=IDUM(I,8)
      JTENSN=IDUM(I,9)
      COSTH=DUM(I,10)
      SINTH=DUM(I,11)
      DO 501 J=1,4
      STRESS(J)=SIGI1(I,J)
      STRAIN(J)=EPSTI1(I,J)
      PSTRAN(J)=EPSPI1(I,J)
      DO 501 K=1,4
  501 CMAT(J,K)=C(I,J,K)
      CALL NCOJL(L,EPSTI,STRAIN,EPSPI,PSTRAN,SIGI,STRESS,
     1COHESN,FRCTN1,SNSWCH,FRCTN2,CRESID,
     2FRESID,MYIELD,IRESID,JTENSN,CMAT,ISTRES,
     3MSWTCH,SBAR,BLAMB,COSTH,SINTH,FMAT,GMAT,
     4SIGNI1,0,SIGNBI)
      IF(MSWTCH.EQ.0) GO TO 400
      IF(LSWTCH.EQ.0) GO TO 300
      DUM(I,1)=COHESN
      DUM(I,2)=FRCTN1
      DUM(I,3)=FRCTN2
      DUM(I,4)=SNSWCH
      DUM(I,5)=CRESID
```

```
      DUM(I,6)=FRESID
      IDUM(I,7)=MYIELD
      IDUM(I,8)=IRESID
      IDUM(I,9)=JTENSN
      DUM(I,10)=COSTH
      DUM(I,11)=SINTH
      GO TO 300
C
  900 WRITE(6,901) NODFEL(I), IPLAST(I)
  901 FORMAT(1H1,15HERROR IN PLASTF//2I5)
      CALL EXIT
  300 IF(LSWTCH.EQ.1) GO TO 301
      DO 12 J=1,8
      FPLAST(J)=0.0
      DO 12 K=1,4
   12 FPLAST(J)=FPLAST(J)+P(I,J,K)*(EPSPI(K)-EPSPI1(I,K))
  301 CONTINUE
      IF(LSWTCH.EQ.0) GO TO 10
      DEPS=0.0
      DO20 J=1,4
      DUMMY=EPSPI(J)-EPSPI1(I,J)
      IF(J.EQ.4)GO TO 21
      DEPS=DEPS+DUMMY*DUMMY
      GO TO 20
   21 DEPS=DEPS+DUMMY*DUMMY/2.0
   20 CONTINUE
      EFFECT(I)=EFFECT(I)+SQRT(2.*DEPS/3.0)
      DO 11 J=1,4
      EPSTI1(I,J)=EPSTI(J)
      EPSPI1(I,J)=EPSPI(J)
   11 SIGI1(I,J)=SIGI(J)
   10 CONTINUE
C
      IF(LSWTCH.EQ.1) GO TO 3
      DO 13 J=1,4
      NODE=NP(I,J)
      K=2*J-1
      N=NODE-NPOUT(ICLUS)
      IF(J.EQ.4) GO TO 14
   16 IF(ITYPE(I,J).EQ.1) GO TO 15
      FNU(N)=FNU(N)+FPLAST(K)
      FNW(N)=FNW(N)+FPLAST(K+1)
      GO TO 13
   15 DUMU= FPLAST(K)*COS(THETA(I,J))+FPLAST(K+1)*SIN(THETA(I,J))
      DUMW=-FPLAST(K)*SIN(THETA(I,J))+FPLAST(K+1)*COS(THETA(I,J))
      FNU(N)=FNU(N)+DUMU
      FNW(N)=FNW(N)+DUMW
      GO TO 13
   14 IF(NODE.NE.0) GO TO 16
   13 CONTINUE
  400 IF(MSWTCH.EQ.1) GO TO 3
      IF(ABS(BLAMB).GT.ALAMB) ALAMB=ABS(BLAMB)
    3 CONTINUE
      IF(LSWTCH.EQ.0) GO TO 17
C
      WRITE(JO) NUMCEL,NELBUF,(NODFEL(K), IPLAST(K),(NP(K,J),
     1ITYPE(K,J),THETA(K,J),J=1,4),((C(K,J,I),I=1,4),(B(K,J,I),I=1,8),
     2J=1,4),((P(K,J,I),J=1,8),EPSTI1(K,I),EPSPI1(K,I),SIGI1(K,I),
     3I=1,4),(DUM(K,I), I=1, 29),EFFECT(K),K=1,NELBUF)
      WRITE(INT) NELBUF,(NODFEL(I),EFFECT(I),(NP(I,J),EPSTI1(I,J),
```

```
     1EPSPI1(I,J),(C(I,J,K),K=1,4),J=1,4),I=1,NELBUF)
      IF((IPRINT.NE.7).AND.(IPRINT.NE.99)) GO TO 17
      IF(ICLUS.NE.1) GO TO 60
      IF(INUM.NE.1) GO TO 60
      WRITE (6,30)
   30 FORMAT(1H1,10X,28HSTRESSES IN PLASTIC ELEMENTS//)
      IF(IFAC.NE.0) GO TO 40
      WRITE(6,35)
   35 FORMAT(5X,16HELASTIC SOLUTION/)
      GO TO 50
   40 WRITE(6,45)IFAC
   45 FORMAT(23H PLASTIC INCREMENT NO.=,I5/)
   50 WRITE (6,55)
   55 FORMAT(12H EL. NUMBER ,8X,12HSIGMAR (PSI),8X,12HSIGMAT (PSI),8X,12
     1HSIGMAZ (PSI),8X,12H  TAU (PSI)/)
   60 WRITE(6,65)(NOOFEL(K),(SIGI1(K,I),I=1,4),K=1,NELBUF)
   65 FORMAT(I7,9X,1P4E20.5)
   17 IF(NUMCEL.LT.NUMELP) RETURN
      REWIND IO
      REWIND JO
      IF(LSWTCH.EQ.1) IPELTP=JO
      RETURN
      END
C
C
C
      SUBROUTINE MISES(I,EPSTI,EPSPI,SIGI,EPSDI,EEFFI,SYI,SMAXI,SI,
     1MSWTCH,BLAMB)
C
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1        NUMEL,ISTRES,NUMPFL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2       KTAPE,KRUN,IPRINT,NUMST,MXSTRT,FUZ(239),
     3                          IPELTP,INT,NPRCDS,IMPBX
      COMMON/A/ U(1600),W(1600),NPOUT(80),NMPCLS(80),FNU(350),
     1        FNW(350)
      COMMON/B/ NADJNP(400),JTYPE(400),SHETA(400),XMASS(400),
     1    SNPUU(400),SNPUW(400),SNPWW(400),FAU(400),FAW(400),
     2    NPADJ(400,8),SADUU(400,8),SADUW(400,8),SADWW(400,8)
C
      DIMENSION BUFF(3280)
      EQUIVALENCE (BUFF,NADJNP)
      DIMENSION NOOFEL(24),IPLAST(24),NP(24,4),ITYPE(24,4),THETA(24,4),
     1C(24,4,4),B(24,4,8),P(24,8,4),EPSTI1(24,4),EPSPI1(24,4),
     2SIGI1(24,4),DUM(24,29),IDUM(24,29)
      EQUIVALENCE (BUFF(1),NOOFEL),(BUFF(25),IPLAST),(BUFF(49),NP),
     1(BUFF(145),ITYPE),(BUFF(241),THETA),(BUFF(337),C),(BUFF(721),B),
     2(BUFF(1489),P),(BUFF(2257),EPSTI1),(BUFF(2353),EPSPI1),
     3(BUFF(2449),SIGI1),(BUFF(2545),DUM),(BUFF(2545),IDUM)
C     DUM(M, 1)=SSTAR(M,1)
C     DUM(M,11)=HSTAR(M,1)
C     DUM(M,21)=NOYILD(M)
C     DUM(M,22)=EPSDI1(M,I)
C     DUM(M,26)=EEFFI1(M)
C     DUM(M,27)=SYI1(M)
C     DUM(M,28)=SMAXI1(M)
C     DUM(M,29)=SI1(M)
      DIMENSION EPSTI(4),EPSPI(4),SIGI(4),EPSDI(4)
      DIMENSION SIGBAR(4),SIGMA(4),EPST(4),EPSD(4),DEPSD(4),SIGMAT(4),
     1ESTAR(10),SSTAR(10),HSTAR(10),EPSDI1(4),DEPSP(4)
      NOYILD=IDUM(I,21)
```

```
      DC 400 J=1,NOYILD
      SSTAR(J)=DUM(I,J)
  400 HSTAR(J)=DUM(I,J+10)
      DC 401 J=1,4
  401 EPSDI1(J)=DJM(I,J+21)
      EEFFI1=DUM(I,26)
      SYI1=DUM(I,27)
      SMAXI1=DJM(I,28)
      SI1=DUM(I,29)
C
      ESTAR(1)=0.0
      IF(NOYILD.EQ.1) GO TO 8
      DC 7 J=2,NOYILD
    7 ESTAR(J)=ESTAR(J-1)+(SSTAR(J)-SSTAR(J-1))/HSTAR(J-1)
    8 DC 1 J=1,4
      SIGBAR(J)=0.0
      DC 1 K=1,4
    1 SIGBAR(J)=SIGBAR(J)+C(I,J,K)*(EPSTI(K)-EPSPI1(I,K))
      SBAR=(ABS(SIGBAR(1)-SIGBAR(2)))**2+(ABS(SIGBAR(1)-SIGBAR(3)))**2
     1    +(ABS(SIGBAR(2)-SIGBAR(3)))**2+6.*(ABS(SIGBAR(4)))**2
      SBAR=SQRT(SBAR/2.)
      IF(MSWTCH.EQ.1) GO TO 3
C
      BLAMB=SBAR/SYI1
      RETURN
    3 IF(SBAR.GE.SYI1    ) GO TO 10
C     CAS  : 1 OR 3, NOW ELASTIC
      IF(SMAXI1   .GE.SYI1    ) GO TO 20
C     CASE =1 ALWAYS WAS ELASTIC AND STILL IS ELASTIC
      ICASE=1
      DC 4 J=1,4
      EPSPI(J)=0.0
      SIGI(J)=SIGBAR(J)
    4 EPSDI(J)=0.0
      EEFFI=0.0
      SYI=SYI1
      SI=SBAR
      IF(SBAR.GT.SMAXI1    ) GO TO 5
      SMAXI=SMAXI1
      GO TO 6
    5 SMAXI=SBAR
    6 RETURN
C     CASE = 3, WAS PREVIOUSLY PLASTIC,NOW ELASTIC
   20 ICASE=3
      DO 21 J=1,4
      EPSPI(J)=EPSPI1(I,J)
      SIGI(J)=SIGBAR(J)
   21 EPSDI(J)=0.0
      EEFFI=EEFFI1
      SYI=SYI1
      SMAXI=SYI
      SI=SBAR
      RETURN
C
   10 IF(SBAR.GT.SYI1    ) GO TO 30
C     CASE = 2 OR 4, JUST AT YIELD STRESS
      IF(SMAXI1    .GE.SYI1    ) GO TO 23
C     CASE = 2, WAS PREVIOUSLY ELASTIC, NOW ON VERGE OF FLOW
      ICASE=2
      GC TO 24
```

```
C     CASE = 4, WAS PREVIOUSLY PLASTIC, NOW ON VERGE OF FLOW
   23 ICASE=4
C
   24 DO 22 J=1,4
      EPSPI(J)=EPSPI1(I,J)
   22 SIGI(J)=SIGBAR(J)
      EPSDI(1)=(2.*SIGBAR(1)-   SIGBAR(2)-   SIGBAR(3))/(2.*SBAR)
      EPSDI(2)=(  -SIGBAR(1)+2.*SIGBAR(2)-   SIGBAR(3))/(2.*SBAR)
      EPSDI(3)=(  -SIGBAR(1)-   SIGBAR(2)+2.*SIGBAR(3))/(2.*SBAR)
      EPSDI(4)=3.*SIGBAR(4)/SBAR
      EEFFI=EEFFI1
      SYI=SYI1
      SMAXI=SYI
      SI=SBAR
      RETURN
C     CASES 5,6 OR 7, PLASTIC FLOW
   30 IF(SMAXI1    .LT.SYI1    ) GO TO 100
C     CASES 5 OR 6
      IF(SI1    .GE.SYI1    ) GO TO 200
C     CASE = 6, WAS PLASTIC, UNLOADED, ELASTIC AT PREVIOUS TIME STEP
      ICASE=6
   32 DUMMY=(SYI1    -SI1    )/(SBAR-SI1    )
      DO 31 J=1,4
      SIGMA(J)=SIGI1(I,J)+DUMMY*(SIGBAR(J)-SIGI1(I,J))
   31 EPST(J)=EPSTI1(I,J)+DUMMY*(EPSTI(J)-EPSTI1(I,J))
      SIGB=(ABS(SIGMA(1)-SIGMA(2)))**2+(ABS(SIGMA(1)-SIGMA(3)))**2
     1    +(ABS(SIGMA(2)-SIGMA(3)))**2+6.*(ABS(SIGMA(4)))**2
      SIGB=SQRT(SIGB/2.)
      EPSD(1)=(2.*SIGMA(1)-   SIGMA(2)-   SIGMA(3))/(2.*SIGB)
      EPSD(2)=(  -SIGMA(1)+2.*SIGMA(2)-   SIGMA(3))/(2.*SIGB)
      EPSD(3)=(  -SIGMA(1)-   SIGMA(2)+2.*SIGMA(3))/(2.*SIGB)
      EPSD(4)=3.*SIGMA(4)/SIGB
      GO TO 201
C     CASE =7, WAS PREVIOUSLY PLASTIC, NOW PLASTIC
  100 ICASE=7
      GO TO 32
C     CASE =5, WAS PREVIOUSLY PLASTIC, NOW FURTHER FLOW
  200 ICASE=5
      DO 33 J=1,4
      SIGMA(J)=SIGI1(I,J)
      EPST(J)=EPSTI1(I,J)
   33 EPSD(J)=EPSDI1(   J)
      SIGB=SYI1
C     COMPUTE NEW PLASTIC STRAINS
  201 IF(ISTRES.NE.2) GO TO 202
C     PLANE STRESS
      XNU=C(I,1,3)/C(I,1,1)
      E=C(I,1,1)*(1.-XNU*XNU)
      A=(7.-13.*XNU+7.*XNU*XNU)/4.+0.75*(2.-5.*XNU+2.*XNU*XNU)*
     1   ((SIGMA(4)/SIGB)*(SIGMA(4)/SIGB)-(SIGMA(1)/SIGB)*(SIGMA(3)/
     2  SIGB))
      A=A/(1.-XNU*XNU)**2
      B=((5.-4.*XNU)/2.)*((SIGBAR(1)/SIGB)*(SIGMA(1)/SIGB)+
     1  (SIGBAR(3)/SIGB)*(SIGMA(3)/SIGB))-((4.-5.*XNU)/2.)*
     2  ((SIGBAR(1)/SIGB)*(SIGMA(3)/SIGB)+(SIGBAR(3)/SIGB)*
     3  (SIGMA(1)/SIGB))+9.*(1.-XNU)*(SIGBAR(4)/SIGB)*(SIGMA(4)/SIGB)
      B=B/(1.-XNU*XNU)
      GO TO 203
C     PLANE STRAIN OR AXISYMMETRIC
  202 XNU=(C(I,1,2)/C(I,1,1))/(1.+C(I,1,2)/C(I,1,1))
```

```
      E=C(I,1,1)*(1.+XNU)*(1.-2.*XNU)/(1.-XNU)
      A=9./(4.*(1.+XNU)**2)
      B=(3./(1.+XNU))*((SIGBAR(1)/SIGB)*EPSD(1)+(SIGBAR(2)/SIGB)*EPSD(2)
     1  +(SIGBAR(3)/SIGB)*EPSD(3)+(SIGBAR(4)/SIGB)*EPSD(4))
C     FIND LOCAL VALUE OF H
  203 IF(NOYILD.GT.1) GO TO 204
      J=1
  205 H=HSTAR(   J)
      GO TO 207
  204 DO 206 K=1,NOYILD
      J=K
      IF(K.EQ.NOYILD) GO TO 205
      IF(SYI1    .LT.SSTAR(   K+1)) GO TO 205
  206 CONTINUE
  207 ALPHA=A-(H/E)**2
      BETA=2.*H/E+B
      GAMMA=(SBAR/SIGB)**2-1.0
      IF(ALPHA.NE.0.0) GO TO 208
      IF(BETA.GT.0.0) GO TO 209
      IERROR=1
  212 WRITE(6,210) T,NOOFEL(I),I
  210 FORMAT(1H1,50HERROR ENCOUNTERED IN PLASTIC FORCES, MISES ROUTINE//
     110H TIME     =,1PE15.5/10H ELEMENT =,I5/10H NUMBER  =,I5)
      WRITE(6,500) (EPSTI1(I,J),J=1,4),(EPSPI1(I,J),J=1,4),
     1(EPSDI1(   J),J=1,4),EEFFI1   ,(EPSTI(J),J=1,4),
     2(EPST(J),J=1,4)                 ,(EPSD(J),J=1,4)
  500 FORMAT(10H EPSTI1  =,1P4E15.5/10H EPSPI1  =,1P4E15.5/
     110H EPSDI1  =,1P4E15.5/10H EEFFI1  =,1PE15.5/10H EPSTI    =,1P4E15
     2.5/10H EPST     =,1P4E15.5                        /10H EPSD     =,
     31P4E15.5)
      WRITE(6,501)(SIGI1(I,J),J=1,4), SYI1    ,SMAXI1    ,SI1     ,
     1(SIGBAR(J),J=1,4),SBAR,(SIGMA(J),J=1,4),SIGB
  501 FORMAT(10H SIGI1   =,1P4E15.5/10H SYI1     =,1PE15.5/
     110H SMAXI1  =,1PE15.5/10H SI1      =,1PE15.5/
     210H SIGBAR  =,1P4E15.5/10H SBAR     =,1PE15.5/
     310H SIGMA   =,1P4E15.5/10H SIGB     =,1PE15.5)
      WRITE(6,502) E,XNU,A,B,H,ALPHA,BETA,GAMMA,IERROR
  502 FORMAT(10H E        =,1PE15.5/10H XNU      =,1PE15.5/
     110H A        =,1PE15.5/10H B        =,1PE15.5/10H H        =,1PE15.5/
     210H ALPHA    =,1PE15.5/10H BETA     =,1PE15.5/10H GAMMA    =,1PE15.5/
     310H IERROR  =,I5)
      CALL EXIT
  209 DELTA=GAMMA/BETA
      GO TO 211
  208 DUMMY=BETA*BETA-4.*ALPHA*GAMMA
  503 IF(DUMMY.GT.0.0) GO TO 213
      DELTA=BETA/(2.*ALPHA)
      IF(DELTA.GT.0.0) GO TO 211
      IERROR=3
      GO TO 212
  213 DUMMY=SQRT(DUMMY)
      DELTA1=(BETA+DUMMY)/(2.*ALPHA)
      DELTA2=(BETA-DUMMY)/(2.*ALPHA)
      IF((DELTA1.GT.0.0).OR.(DELTA2.GT.0.0)) GO TO 600
      IERROR=4
      GO TO 212
  600 IF((DELTA1.GT.0.0).AND.(DELTA2.GT.0.0)) GO TO 601
      IF(DELTA1.GT.0.0) DELTA=DELTA1
      IF(DELTA2.GT.0.0) DELTA=DELTA2
      GO TO 211
```

```
  601 IF(DELTA1.GE.DELTA2) DELTA=DELTA2
      IF(DELTA1.LT.DELTA2) DELTA=DELTA1
  211 DEEFF=SIGB*DELTA/E
C
      IC=1
      DE=0.0
      S=SYI1
      DEEFFL=0.0
      DEEFFH=0.0
      SIGIBM=SBAR-SYI1
      DEEFFM=0.0
      ISW=1
  215 DO 216 J=1,4
      SIGI(J)=0.0
      DO 216 K=1,4
  216 SIGI(J)=SIGI(J)+C(I,J,K)*(EPSTI(K)-EPSPI1(I,K)-DEEFF*EPSD   (K))
      SIGIB=(ABS(SIGI(1)-SIGI(2)))**2 +(ABS(SIGI(2)-SIGI(3)))**2
     1      +(ABS(SIGI(3)-SIGI(1)))**2+6.*(ABS(SIGI(4)))**2
      SIGIB=SQRT(SIGIB/2.)
      SOI=S+H*(DEEFF-DE)
      IF(ISW .EQ. 2) GO TO 217
      IF(SIGIB-SOI .GT. SIGIBM) GO TO 227
      SIGIBM=SIGIB-SOI
      DEEFFM=DEEFF
  227 CONTINUE
  226 IF(ABS(SOI-SIGIB).LE.0.01*SOI) GO TO 217
      IF(DEEFFH.EQ.0.0) GO TO 218
      IF(SOI.GT.SIGIB)  GO TO 219
      DEEFFL=DEEFF
  220 DEEFF=(DEEFFL+DEEFFH)/2.
      GO TO 215
  219 DEEFFH=DEEFF
      GO TO 220
  218 IF(SOI.GT.SIGIB) GO TO 219
      DEEFFL=DEEFF
      DEEFF=2.*DEEFF
      IC=IC+1
      IF(IC.LE.20) GO TO 215
      DEEFF=DEEFFM
      ISW=2
      GO TO 215
  217 DO 221 J=1,4
  221 SIGI(J)=SIGI(J)*SOI/SIGIB
      EEFFI=EEFFI1+DEEFF
      IF(NOYILD.EQ.1) GO TO 222
      IF(EEFFI1.GE.ESTAR(NOYILD)) GO TO 222
      J=1
  224 IF(EEFFI1.LT.ESTAR(J)) GO TO 223
      J=J+1
      GO TO 224
  223 IF(EEFFI.LE.ESTAR(J)) GO TO 222
      S=SSTAR(J)
      H=HSTAR(J)
      DE=ESTAR(J)-EEFFI1
      DEEFFL=DEEFF
      DEEFF=2.*DEEFF
      IC=1
      DEEFFH=0.0
      ISW=1
      GO TO 215
```

```
  222 DO 225  J=1,4
  225 EPSPI(J)=EPSPI1(I,J)+DEEFF*EPSD(J)
      EPSDI(1)=(2.*SIGI(1)-    SIGI(2)-    SIGI(3))/(2.*SOI)
      EPSDI(2)=(  -SIGI(1)+2.*SIGI(2)-    SIGI(3))/(2.*SOI)
      EPSDI(3)=(  -SIGI(1)-    SIGI(2)+2.*SIGI(3))/(2.*SOI)
      EPSDI(4)=3.*SIGI(4)/SOI
      SYI=SOI
      SMAXI=SOI
      SI=SOI
      RETURN
      END
C
C
C
      SUBROUTINE COULMR(I,EPSTI,EPSPI,SIGI,KORNER,FYLDI,EPSDI,MSWTCH,
     1BLAMBI
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1       NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,CLAMB,
     2      KTAPE,KRUN,IPRINT,NUMST,MXSTRT,FUZ(239),
     3                          IPELTP,INT,NPRCDS,IMPBX
      COMMON/A/ U(1600),W(1600),NPOUT(80),NMPCLS(80),FNU(350),
     1     FNW(350)
      COMMON/B/NADJNP(400),JTYPE(400),SHETA(400),XMASS(400),
     1   SNPUU(400),SNPUW(400),SNPWW(400),FAU(400),FAW(400),
     2   NPADJ(400,8),SADUU(400,8),SADUW(400,8),SADWW(400,8)
      DIMENSION BUFF(3280)
      EQUIVALENCE (BUFF,NADJNP)
      DIMENSION NOOFEL(24),IPLAST(24),NP(24,4),ITYPE(24,4),THETA(24,4),
     1C(24,4,4),B(24,4,8),P(24,8,4),EPSTI1(24,4),EPSPI1(24,4),
     2SIGI1(24,4),DUM(24,29),IDUM(24,29)
      EQUIVALENCE (BUFF(1),NOOFEL),(BUFF(25),IPLAST),(BUFF(49),NP),
     1(BUFF(145),ITYPE),(BUFF(241),THETA),(BUFF(337),C),(BUFF(721),B),
     2(BUFF(1489),P),(BUFF(2257),EPSTI1),(BUFF(2353),EPSPI1),
     3(BUFF(2449),SIGI1),(BUFF(2545),DUM),(BUFF(2545),IDUM)
C      DUM(M,3)=COSTH
C     DUM(M, 4)=KORNER
C     DUM(M, 5)=FYLDI1
C     DUM(M, 6)=EPSDI1
      DIMENSION EPSTI(4),EPSPI(4),SIGI(4),EPSDI(4)
      DIMENSION EPSDI1(4),SIGBAR(4),SIGMA(4),EPST(4),EPSD(4),ED(4),SX(4)
C
C     SIMPLE COULOMB MOHR YIELD CONDITION
      ALPHA=DUM(I,1)
      CAPPA=DUM(I,2)
      COSTH=DUM(I,3)
      KORNER=IDUM(I,4)
      FYLDI1=DUM(I,5)
      DO 1  J=1,4
    1 EPSDI1(J)=DUM(I,J+5)
C
      DO 2  J=1,4
      SIGBAR(J)=0.0
      DO 2  K=1,4
    2 SIGBAR(J)=SIGBAR(J)+C(I,J,K)*(EPSTI(K)-EPSPI1(I,K))
      AI1BAR=SIGBAR(1)+SIGBAR(2)+SIGBAR(3)
      AI2BAR=((SIGBAR(1)-SIGBAR(2))*(SIGBAR(1)-SIGBAR(2))
     1        +(SIGBAR(2)-SIGBAR(3))*(SIGBAR(2)-SIGBAR(3))
     2        +(SIGBAR(3)-SIGBAR(1))*(SIGBAR(3)-SIGBAR(1)))/6.0
     3        + SIGBAR(4)*SIGBAR(4)
      IF(AI2BAR.LE.0.0) GO TO 3
```

```
      FYLDBR=ALPHA*AI1BAR+SQRT(AI2BAR)
      GO TO 4
    3 FYLDBR=ALPHA*AI1BAR
    4 IF(MSWTCH.EQ.1) GO TO 33
      IF(CAPPA.EQ.0.0)GO TO 65
      BLAMB=FYLDBR /CAPPA
      RETURN
   65 BLAMB=1.E+38
      RETURN
   33 IF(FYLDBR.GE.CAPPA) GO TO 6
      KORNER=0
      DO 5  J=1,4
      SIGI(J)=SIGBAR(J)
      EPSDI(J)=0.0
    5 EPSPI(J)=EPSPI1(I,J)
      FYLDI=FYLDBR
      RETURN
    6 IF(FYLDBR.GT.CAPPA) GO TO 10
      DO 7  J=1,4
      SIGI(J)=SIGBAR(J)
    7 EPSPI(J)=EPSPI1(I,J)
      FYLDI=FYLDBR
      IF(AI2BAR.LT.0.001) GO TO 8
      KORNER=0
      DUM1=SQRT(AI2BAR)
   64 EPSDI(1)=ALPHA+(2.*SIGI(1)-     SIGI(2)-     SIGI(3))/(6.*DUM1)
      EPSDI(2)=ALPHA+(   -SIGI(1)+2.0*SIGI(2)-     SIGI(3))/(6.*DUM1)
      EPSDI(3)=ALPHA+(   -SIGI(1)-     SIGI(2)+2.0*SIGI(3))/(6.*DUM1)
      EPSDI(4)=SIGI(4)/DUM1
      CALL LGTH(EPSDI)
      RETURN
    8 KORNER=1
      DO 9  J=1,4
    9 EPSDI(J)=0.0
      RETURN
   10 IF(FYLDI1.LT.CAPPA) GO TO 12
      DO 11  J=1,4
      SIGMA(J)=SIGI1(I,J)
      EPST(J)=EPSTI1(I,J)
   11 EPSD(J)=EPSDI1(J)
      GO TO 16
   12 DUMMY=(CAPPA-FYLDI1)/(FYLDBR-FYLDI1)
      DO 13  J=1,4
      SIGMA(J)=SIGI1(I,J)+DUMMY*(SIGBAR(J)-SIGI1(I,J))
   13 EPST(J)=EPSTI1(I,J)+DUMMY*(EPSTI(J)-EPSTI1(I,J))
      AI2=((SIGMA(1)-SIGMA(2))*(SIGMA(1)-SIGMA(2))
     1    +(SIGMA(2)-SIGMA(3))*(SIGMA(2)-SIGMA(3))
     2    +(SIGMA(3)-SIGMA(1))*(SIGMA(3)-SIGMA(1)))/6.0
     3    +SIGMA(4)*SIGMA(4)
      IF(AI2.LE.1.0E-5) GO TO 14
      DUMMY=SQRT(AI2)
      KORNER=0
      EPSD(1)=ALPHA+(2.*SIGMA(1)-    SIGMA(2)-    SIGMA(3))/(6.*DUMMY)
      EPSD(2)=ALPHA+(   -SIGMA(1)+2.*SIGMA(2)-    SIGMA(3))/(6.*DUMMY)
      EPSD(3)=ALPHA+(   -SIGMA(1)-    SIGMA(2)+2.*SIGMA(3))/(6.*DUMMY)
      EPSD(4)=SIGMA(4)/DUMMY
      CALL LGTH(EPSD)
      GO TO 16
   14 KORNER=1
      DO 15  J=1,4
```

```
 15 EPSD(J)=0.0
 16 IF(KORNER.EQ.1) GO TO 28
 17 AX=C(I,1,1)*EPSD(1)+C(I,1,2)*EPSD(2)+C(I,1,3)*EPSD(3)
    AY=C(I,2,1)*EPSD(1)+C(I,2,2)*EPSD(2)+C(I,2,3)*EPSD(3)
    AZ=C(I,3,1)*EPSD(1)+C(I,3,2)*EPSD(2)+C(I,3,3)*EPSD(3)
    AW=C(I,4,4)*EPSD(4)
    B1=AX+AY+AZ
    B2=((SIGBAR(1)-SIGBAR(2))*(AX-AY)+(SIGBAR(2)-SIGBAR(3))*(AY-AZ)
   1   +(SIGBAR(3)-SIGBAR(1))*(AZ-AX))/6.0+(SIGBAR(4)*AW)
    B3=(ABS(AX-AY))**2+(ABS(AY-AZ))**2+(ABS(AZ-AX))**2
    B3=B3/6.0+AW*AW
    D1=B3-(ALPHA*B1)*(ALPHA*B1)
    D2=2.*(ALPHA*ALPHA*AI1BAR*B1-ALPHA*CAPPA*B1-B2)
    D3=AI2BAR-CAPPA*CAPPA+2.*ALPHA*CAPPA*AI1BAR-ALPHA*ALPHA
   1   *AI1BAR*AI1BAR
    IF(D1.NE.0.0) GO TO 18
    ALAMB=-D3/D2
    IF(ALAMB.GT.0.0) GO TO 24
    IERROR=1
    ALAMB1=0.0
    ALAMB2=0.0
    DUMMY =0.0
    GO TO 23
 18 DUMMY=D2*D2-4.*D1*D3
    IF(DUMMY.GE.0.0) GO TO 19
    IF(ABS(DUMMY/(D2*D2)).LT.0.015) GO TO 19
    IERROR=2
    ALAMB=0.0
    ALAMB1=0.0
    ALAMB2=0.0
    GO TO 23
 19 IF(DUMMY.GT.0.0) GO TO 20
    ALAMB=-D2/(2.*D1)
    IF(ALAMB.GT.0.0) GO TO 24
    IERROR=3
    ALAMB1=0.0
    ALAMB2=0.0
    GO TO 23
 20 ALAMB1=(-D2+SQRT(DUMMY))/(2.*D1)
    ALAMB2=(-D2-SQRT(DUMMY))/(2.*D1)
    IF((ALAMB1.GT.0.0).OR.(ALAMB2.GT.0.0)) GO TO 21
    IERROR=4
    GO TO 23
 21 IF((ALAMB1.GT.0.0).AND.(ALAMB2.GT.0.0)) GO TO 22
    IF(ALAMB1.GT.0.0) ALAMB=ALAMB1
    IF(ALAMB2.GT.0.0) ALAMB=ALAMB2
    GO TO 24
 22 IF(ALAMB1.GE.ALAMB2) ALAMB=ALAMB2
    IF(ALAMB1.LT.ALAMB2) ALAMB=ALAMB1
    GO TO 24
 23 WRITE(6,100) T,NOOFEL(I),I
100 FORMAT(1H1,45HERROR IN PLASTIC FORCES, COULOMB-MOHR ROUTINE//
   110H TIME     =,1PE15.5/10H ELEMENT =,I5/10H NUMBER  =,I5)
    WRITE(6,101) (EPSTI1(I,J),J=1,4),(EPSPI1(I,J),J=1,4),
   1(EPSDI1(J),J=1,4),(EPSTI(J),J=1,4),(EPST(J),J=1,4),
   2(EPSD(J),J=1,4)
101 FORMAT(10H EPSTI1  =,1P4E15.5/10H EPSPI1  =,1P4E15.5/
   110H EPSDI1  =,1P4E15.5/10H EPSTI   =,1P4E15.5/
   210H EPST    =,1P4E15.5/10H EPSD    =,1P4E15.5)
    WRITE(6,102) (SIGI1(I,J),J=1,4),(SIGBAR(J),J=1,4),(SIGMA(J),J=1,4)
```

```
102 FORMAT(10H SIGIL    =,1P4E15.5/10H SIGBAR   =,1P4E15.5/
   110H SIGMA    =,1P4E15.5)
    WRITE(6,103) ALPHA,CAPPA,COSTH,KORNER,FYLDI1,FYLDBR,AI1BAR,
   1AI2BAR
103 FORMAT(10H ALPHA    =,1PE15.5/10H KAPPA    =,1PE15.5/10H COS(TH) =,
   11PE15.5/10H KORNER   =,I5/10H FYLDI1   =,1PE15.5/10H FYLDBR   =,
   21PE15.5/10H I1BAR    =,1PE15.5/10H I2BAR    =,1PE15.5)
    WRITE(6,104) C(I,1,1),C(I,1,2),C(I,1,3),C(I,4,4)
104 FORMAT(10H C(1,1)   =,1PE15.5/10H C(1,2)   =,1PE15.5/10H C(1,3)   =,
   11PE15.5/10H C(4,4)   =,1PE15.5)
    WRITE(6,105) AX,AY,AZ,AW,B1,B2,B3,D1,D2,D3
105 FORMAT(10H AX       =,1PE15.5/10H AY       =,1PE15.5/10H AZ       =,
   11PE15.5/10H AW       =,1PE15.5/10H B1       =,1PE15.5/10H B2       =,
   21PE15.5/10H B3       =,1PF15.5/10H D1       =,1PE15.5/10H D2       =,
   31PE15.5/10H D3       =,1PE15.5)
    WRITE(6,106) ALAMB,ALAMB1,ALAMB2,DUMMY,IERROR
106 FORMAT(10H ALAMB    =,1PE15.5/10H ALAMB1   =,1PE15.5/10H ALAMB2   =,
   11PE15.5/10H DUMMY    =,1PE15.5/10H IERROR   =,I5)
    CALL EXIT
 24 ALAMBL=0.0
    ALAMBH=0.0
    IC=1
 25 DO 26 J=1,4
    SIGI(J)=0.0
    DO 26 K=1,4
 26 SIGI(J)=SIGI(J)+C(I,J,K)*(EPSTI(K)-EPSPI1(I,K)-ALAMB*EPSD(K))
    AI1=SIGI(1)+SIGI(2)+SIGI(3)
    AI2=((ABS(SIGI(1)-SIGI(2)))**2+(ABS(SIGI(2)-SIGI(3)))**2
   1     +(ABS(SIGI(3)-SIGI(1)))**2)/6.0+SIGI(4)*SIGI(4)
    FSTAR=ALPHA*AI1+SQRT(AI2)-CAPPA
    IF(FSTAR.NE.0.0) GO TO 53
    IF(AI2.GT.0.001) GO TO 51
 27 KORNER=1
    DO 50 J=1,4
 50 EPSPI(J)=EPSPI1(I,J)+ALAMB*EPSD(J)
    GO TO 36
 51 KORNER=0
    FYLDI=CAPPA
    EPSDI(1)=ALPHA+(2.*SIGI(1)-    SIGI(2)-    SIGI(3))/(6.*SQRT(AI2))
    EPSDI(2)=ALPHA+(   -SIGI(1)+2.*SIGI(2)-    SIGI(3))/(6.*SQRT(AI2))
    EPSDI(3)=ALPHA+(   -SIGI(1)-    SIGI(2)+2.*SIGI(3))/(6.*SQRT(AI2))
    EPSDI(4)=SIGI(4)/SQRT(AI2)
    CALL LGTH(EPSDI)
    DO 52 J=1,4
 52 EPSPI(J)=EPSPI1(I,J)+ALAMB*EPSD(J)
    RETURN
 53 ABAR2=(ABS(CAPPA/ALPHA-AI1))**2/3.
    BBAR2=2.*AI2
    RBAR2=ABAR2+BBAR2
    IF(RBAR2.LE.0.001) GO TO 27
    IF((CAPPA/ALPHA-AI1).LE.0.0) GO TO 27
    AI2S=(CAPPA-ALPHA*AI1)**2
    BBARST=SQRT(2.*AI2S)
    IF(BBAR2.GT.0.0) GO TO 54
    ALAMBH=ALAMB
    GO TO 57
 54 BBAR=SQRT(BBAR2)
    IF(ABS(BBARST-BBAR).LE.0.01*BBARST) GO TO 61
C=====================================*******************************
    PRINT 999,I,NCUFFL(I),IC,ALAMB,BBAR,BBARST
```

```
 999  FORMAT(3I10,3E15.5)
C========================================*********************************
      IC=IC+1
      IF(IC.LE.20) GO TO 55
      IERROR=8
      ALAMB1=BBARST
      ALAMB2=BBAR
      DUMMY=AI2
      GO TO 23
   55 IF(ALAMBH.GT.0.0) GO TO 58
      IF(BBAR.LT.BBARST) GO TO 56
      ALAMBL=ALAMB
      ALAMB=2.*ALAMB
      GO TO 25
   56 ALAMBH=ALAMB
   57 ICOUNT=I
      GO TO 60
   58 IF(BBAR.GT.BBARST) GO TO 59
      ALAMBH=ALAMB
      GO TO 60
   59 ALAMBL=ALAMB
   60 ALAMB=(ALAMBH+ALAMBL)/2.
      GO TO 25
   61 S=AI1/3.
      DO 62 J=1,3
   62 SX(J)=SIGI(J)-S
      SX(4)=SIGI(4)
      DUMMY=BBARST/BBAR
      DO 63 J=1,3
      SX(J)=DUMMY*SX(J)
   63 SIGI(J)=S+SX(J)
      SIGI(4)=DUMMY*SX(4)
      GO TO 51
   28 XNU=C(I,1,2)/(C(I,1,1)+C(I,1,2))
      EBAR=C(I,1,1)/(1.-XNU)
      EMOD=EBAR*(1.+XNU)*(1.-2.*XNU)
      DO 29 J=1,3
   29 EPSD(J)=EPSTI(J)-(CAPPA*(1.-2.*XNU)/(3.*ALPHA*EMOD))-EPSPI1(I,J)
      EPSD(4)=EPSTI(4)-EPSPI1(I,4)
      CALL LGTH(EPSD)
      ALGTH=(ABS(EPSD(1)))**2+(ABS(EPSD(2)))**2+(ABS(EPSD(3)))**2
      ALGTH=SQRT(ALGTH)
      DELTAD=EPSD(1)+EPSD(2)+EPSD(3)
      COSTHB=DELTAD/(SQRT(3.)*ALGTH)
      IF(ALGTH.EQ.0.0) COSTHB=1.0
      IF(COSTHB.GE.COSTH) GO TO 42
C
      ABAR2=(ABS(CAPPA/ALPHA-AI1BAR))**2/3.
      BBAR2=2.*AI2BAR
      RBAR2=ABAR2+BBAR2
      IF(RBAR2.LT.1.0E-5) GO TO 42
      IF((CAPPA/ALPHA-AI1BAR).LE.0.0) GO TO 42
      S=AI1BAR/3.
      AI2S=(CAPPA-ALPHA*AI1BAR)**2
      BBARST=SQRT(2.*AI2S)
      BBAR=SQRT(BBAR2)
      DO 30 J=1,3
   30 SX(J)=(BBARST/BBAR)*(SIGBAR(J)-S)
      SX(4)=(BBARST/BBAR)*SIGBAR(4)
      DO 31 J=1,3
```

```
   31 SIGI(J)=SX(J)+S
      SIGI(4)=SX(4)
      ED(1)=(SIGI(1)-XNU*(SIGI(2)+SIGI(3)))/EMOD
      ED(2)=(SIGI(2)-XNU*(SIGI(1)+SIGI(3)))/EMOD
      ED(3)=(SIGI(3)-XNU*(SIGI(1)+SIGI(2)))/EMOD
      ED(4)=2.*(1.+XNJ)*SIGI(4)/EMOD
      DO 32 J=1,4
   32 EPSPI(J)=EPSTI(J)-ED(J)
      KORNER=0
      FYLDI=CAPPA
      DUM1=SQRT(AI2S)
      GO TO 64
   42 DO 43 J=1,3
   43 EPSPI(J)=EPSTI(J)-(CAPPA*(1.-2.*XNU)/(3.*ALPHA*EMOD))
      EPSPI(4)=EPSTI(4)
   36 S=CAPPA/(3.*ALPHA)
      DO 37 J=1,3
   37 SIGI(J)=S
      SIGI(4)=0.0
      KORNER=1
      FYLDI=CAPPA
      DO 38 J=1,4
   38 EPSDI(J)=0.0
      RETURN
      END
C
C
C
      SUBROUTINE LGTH(E)
      DIMENSION E(4),A(4)
      DO 1 I=1,4
    1 A(I)=ABS(E(I))
      B=AMAX1(A(1),A(2),A(3),A(4))
      DO 2 I=1,4
    2 E(I)=E(I)/B
      RETURN
      END
C
C
C
      SUBROUTINE NCOJL(NJME,EPSTI,EPSTI1,EPSPI,EPSPI1,SIGI,SIGI1,
     1COHESN,FRCTN1,SNSWCH,FRCTN2,CRESID,
     2FRESID,MYIELD,IRESID,JTENSN,CMAT,ISTRES,
     3MSWTCH,SBAR,BLAMB,COSTH,SINTH,F,G,
     4SIGNI1,NSWTCH,SIGNBI)
      DIMENSION CMAT(4,4),EPSTI(4),EPSTI1(4),EPSPI(4),EPSPI1(4),
     1 SIGI(4),SIGI1(4)
      DIMENSION F(4,4),G(4,4),SIGNI(4),SIGNI1(4),SIGNBI(4),EPTNI(4),
     1 EPTNI1(4)
C     MSWTCH=0 COMPUTE BLAMB ONLY
C     NSWTCH=1 COMPUTE F,G, AND SIGNI1 ONLY
      DO 1 I=1,4
      DO 1 J=1,4
      F(I,J)=0.0
    1 G(I,J)=0.0
      F(1,1)=COSTH*COSTH
      F(3,1)=SINTH*SINTH
      F(4,1)=SINTH*COSTH
      F(2,2)=1.0
      F(1,3)=F(3,1)
```

```
      F(3,3)=F(1,1)
      F(4,3)=-F(4,1)
      F(1,4)=-2.*F(4,1)
      F(3,4)=-F(1,4)
      F(4,4)=F(1,1)-F(3,1)
      DO 2 I=1,4
      DO 2 J=1,4
    2 G(I,J)=F(I,J)
      G(4,1)=-G(4,1)
      G(4,3)=-G(4,3)
      G(1,4)=-G(1,4)
      G(3,4)=-G(3,4)
C
      DO 3 I=1,4
      SIGNI1(I)=0.0
      DO 3 J=1,4
    3 SIGNI1(I)=SIGNI1(I)+F(I,J)*SIGI1(J)
      IF(NSWTCH.EQ.1) RETURN
      DO 4 I=1,4
      EPTNI1(I)=0.0
      EPTNI (I)=0.0
      DO 4 J=1,4
      EPTNI1(I)=EPTNI1(I)+G(J,I)*EPSTI1(J)
    4 EPTNI (I)=EPTNI (I)+G(J,I)*EPSTI (J)
      DO 5 I=1.4
      SIGNBI(I)=SIGNI1(I)
      DO 5 J=1,4
    5 SIGNBI(I)=SIGNBI(I)+CMAT(I,J)*(EPTNI(J)-EPTNI1(J))
      SBAR=SIGNBI(4)
C
      TNPHI1=TAN(FRCTN1)
      TNPHI2=TAN(FRCTN2)
      TAUNT=ABS(SIGNBI(4))
       SNI=SIGNBI(3)
      IF(MSWTCH.EQ.1) GO TO 10
      IF(SNI.GT.0.0.AND.JTENSN.EQ.0) GO TO 6
      AUMER=TAUNT+SNI*TNPHI1
      DENOM=COHESN
      IF(DENOM.GT.0.0) GO TO 7
      IF(AUMER.GT.0.0) GO TO 6
      IF(FRCTN1.GT.FRCTN2) GO TO 50
      BLAMB=0.0
      GO TO 9
    6 BLAMB=1.0E+38
      GO TO 9
    7 IF(AUMER.GT.0.0) GO TO 8
      AUMER=TAUNT+SNI*TNPHI2
      IF(AUMER.GT.0.0) GO TO 8
      BLAMB=0.0
      GO TO 9
    8 BLAMB=AUMER/DENOM
      IF(FRCTN1.EQ.FRCTN2) RETURN
      SIGFF=SNI/BLAMB
      IF(SIGFF.GE.SNSWCH) RETURN
   50 AUMER=TAUNT+SNI*TNPHI2
      DENOM=COHESN-SNSWCH*(TNPHI1-TNPHI2)
      BLAMB=AUMER/DENOM
      IF(AUMER.LE.0.0) BLAMB=0.0
    9 RETURN
C
```

```
10    IF(JTENSN.EQ.1) GO TO 100
      IF(SNI.GE.0.0) GO TO 150
      GO TO 105
100   IF(TNPHI1.LE.0.0) GO TO 105
      IF(SNI.LT.COHESN/TNPHI1) GO TO 105
      IF(IRESID.EQ.0) GO TO 160
      COHESN=CRESID
      FRCTN1=FRESID
      FRCTN2=FRESID
      GO TO 160
105   IF(MYIELD.EQ.1.AND.IRESID.EQ.1) GO TO 110
      IF(SNI.LT.SNSWCH) GO TO 120
110   TAUNTB=COHESN-SNI*TNPHI1
      GO TO 130
120   TAUNTB=COHESN-SNSWCH*TNPHI1-(SNI-SNSWCH)*TNPHI2
130   IF(TAUNT.LT.TAUNTB) GO TO 170
      IF(MYIELD.EQ.1) GO TO 140
      IF(IRESID.EQ.0) GO TO 140
      COHESN=CRESID
      FRCTN1=FRESID
      FRCTN2=FRESID
      IF(FRESID.GT.0.) GO TO 132
      IF(SNI.GE.0.0) GO TO 150
      GO TO 135
132   IF(SNI.LT.COHESN/TAN(FRESID)) GO TO 135
      GO TO 160
135   TAUNTB=COHESN - SNI*TAN(FRESID)
140   SIGNI(4)=SIGNBI(4)*TAUNTB/TAUNT
      SIGNI(3)=SIGNBI(3)
      SIGNI(2)=SIGNBI(2)
      SIGNI(1)=SIGNBI(1)
      MYIELD=1
      GO TO 15
150   SIGNI(4)=0.0
      SIGNI(3)=0.0
      SIGNI(2)=SIGNBI(2)
      SIGNI(1)=SIGNBI(1)
      MYIELD=1
      GO TO 15
160   SIGNI(4)=0.0
      SIGNI(3)=COHESN/TAN(FRCTN1)
      SIGNI(2)=SIGNBI(2)
      SIGNI(1)=SIGNBI(1)
      MYIELD=1
      GO TO 15
170   DO 180 I=1,4
180   SIGNI(I)=SIGNBI(I)
C
   15 DO 16 I=1,4
      SIGI(I)=0.0
      DO 16 J=1,4
   16 SIGI(I)=SIGI(I)+G(I,J)*SIGNI(J)
C
      IF(ISTRES.EQ.2) GO TO 17
      XNU=CMAT(1,3)/(CMAT(1,1)+CMAT(1,3))
      EBAR=CMAT(1,1)/(1.-XNU)
      EMOD=EBAR*(1.+XNU)*(1.-2.*XNU)
      GO TO 18
   17 XNU=CMAT(1,3)/CMAT(1,1)
      EMOD=CMAT(1,1)*(1.-XNU*XNU)
```

```
   18 EPSPI(1)=(SIGI(1)-XNU*(SIGI(2)+SIGI(3)))/EMOD
      EPSPI(2)=(SIGI(2)-XNU*(SIGI(3)+SIGI(1)))/EMOD
      EPSPI(3)=(SIGI(3)-XNU*(SIGI(1)+SIGI(2)))/EMOD
      EPSPI(4)=SIGI(4)/CMAT(4,4)
      DO 19 I=1,4
   19 EPSPI(I)=EPSTI(I)-EPSPI(I)
C
      RETURN
      END
C
C
C
      OVERLAY(MOHAN,14,0 )
      PROGRAM LNK3
C
      COMMON MAXNP,MXCLS,MXADJP,MXZONE,MXNPB,NZONES,MXPELB,NUMNP,
     1        NUMEL,ISTRES,NUMPEL,NUMELP,PERIOD,NMKCLS,FACTOR,ALAMB,
     2       KTAPE,KRUN,IPRINT,NUMST,MXSTRT,FUZ(239),
     3                    IPELTP,INT,NPRCDS,IMPBX
C
      COMMON/A/ U(1600),W(1600),NPOUT(80),NMPCLS(80),FNU(350),
     1       FNW(350)
C
      DIMENSION STNPU(4,350),STNPW(4,350),STADU(4,350,8),
     1STADW(4,350,8),NADJNP(350),NADJEL(350),NPADJ(350,8)
C
      DIMENSION ITYPE(350),SHETA(350),XMASS(350),SNPLU(350),SNPUW(350),
     1SNPWW(350),FAU(350),FAW(350),SADUU(350,8),SADUW(350,8),
     2SADWW(350,8)
C
      DIMENSION NPTN(1600),COM(16),SIG(350,4),EPST(24,4),EFFECT(24),
     1SIGPL(350,4),SIGMX(350),SIGMN(350),THETA(350),NOOFEL(24),
     2NP(24,4),EPSP(24,4),C(24,4,4),SIGMAP(4),EPSE(4),SIGMA(4)
C
      EQUIVALENCE (SIGMX,FNU),(SIGMN,FNW),(MAXNP,COM(1))
C
      EQUIVALENCE (STNPU(1),ITYPE),(STNPU(351),SHETA),(STNPU(701),XMASS)
     1,(STNPU(1021),SNPUJ),(STNPW(1),SNPUW),(STNPW(351),SNPWW),
     2(STNPW(701),FAU),(STNPW(1051),FAW),(STADU(1),SADUU),
     3(STADU(2801),SADUW),(STADU(5601),SADWW)
      FQUIVALENCE(FUZ(1),NOOFEL),(FUZ(25),NP),(FUZ(121),EPSP)
C
      MOHAN=5HMOHAN
      PI=3.1415927
      ISWTCH=0
      IF((NUMNP.LE.MXNPB).AND.(NUMPEL.EQ.0)) ISWTCH=1
      REWIND 10
      REWIND INT
      REWIND 1
C*********************************************************************
      IF(IMPBX.NE.1) GO TO 1
      REWIND 15
      READ(15)DUMMY,DUMMY,(DUMMY1,DUMMY2,I= 1,NUMNP),(DUMMY1,DUMMY2,
     1DUMMY3,DUMMY4,DUMMY5,I=1,NUMEL)
C*********************************************************************
    1 READ(10)  N1,N2,N3,N4,N5,N6,N7,(NADJNP(I),ITYPE(I),SHETA(I),
     1XMASS(I),SNPUJ(I),SNPUW(I),SNPWW(I),FAU(I),FAW(I),
     2(NPADJ(I,J),SADUU(I,J),SADUW(I,J),SADWW(I,J),J=1,MXADJP),I=1,N7)
      IF(N4.LT.NUMNP) GO TO 1
      READ(10) (NPTN(I),I=1,NUMNP)
```

```
C
      IC=1
    2 READ(10) NPLOW,NPHIGH,NPOUT(IC),NUMCP,N5,NMPCLS(IC),N7,(NADJNP(I),
     1 NADJEL(I),(NPADJ(I,J),J=1,MXADJP),(STNPU(K,I),STNPW(K,I),
     2K=1,4),((STADJ(K,I,J),STADW(K,I,J),K=1,4),J=1,MXADJP),I=1,N7)
      IF(ISWTCH.EQ.1) GO TO 30
      WRITE(1) NPLOW,NPHIGH,NPOUT(IC),NUMCP,N5,NMPCLS(IC),N7,(NADJNP(I),
     1 NADJEL(I),(NPADJ(I,J),J=1,MXADJP),(STNPU(K,I),STNPW(K,I),
     2K=1,4),((STADJ(K,I,J),STADW(K,I,J),K=1,4),J=1,MXADJP),I=1,N7)
      IC=IC+1
      IF(NUMCP.LT.NUMNP) GO TO 2
   30 REWIND 10
      REWIND 1
      READ(INT)(COM(I),I=1,16)
C
      IF(IPELTP.EQ.12) IOUTAP=3
      IF(IPELTP.EQ. 3) IOUTAP=12
      WRITE(6,3) IOUTAP
    3 FORMAT(1H1,26HOUTPUT HISTORY TAPE IS NO.,I5//)
C
      REWIND IOUTAP
      WRITE(IOUTAP)(COM(I),I=1,16),NPRCDS,(NPTN(I),I=1,NUMNP)
C
      NPRCDS=NPRCDS+1
      DO 100 IPRCDS=1,NPRCDS
C
      DO 200 I=1,NUMNP
      READ(INT) UBAR,WBAR
      IF(IPRCDS.GT.1) GO TO 201
      U(I)=UBAR
      W(I)=WBAR
      GO TO 200
  201 U(I)=U(I)+UBAR
      W(I)=W(I)+WBAR
  200 CONTINUE
C
      DO 101 ICLUS=1,NMKCLS
C
      DO 4 I=1,MXNPB
      DO 4 J=1,4
    4 SIGPL(I,J)=0.0
C
      IF(IPRCDS.GT.1) GO TO 104
      WRITE(6,25)
   25 FORMAT(1H1,16HELASTIC SOLUTION//)
      IF(NUMPEL .EQ.0) GO TO 102
      GO TO 106
C
  104 NUM=IPRCDS-1
      WRITE(6,5) NUM
    5 FORMAT(1H1,22HPLASTIC INCREMENT NO.=,I5//)
C
  106 CONTINUE
      NUM=NMPCLS(ICLUS)
      IF(NUM.EQ.0) GO TO 102
      DO 6 II=1,NUM
C
      READ(INT)NUMELB,(NOOFEL(I),EFFECT(I),(NP(I,J),EPST(I,J),EPSP(I,J),
     1(C(I,J,K),K=1,4),J=1,4),I=1,NUMELB)
C
```

```
      DO 7 I=1,NUMELB
      DC 32 J=1,4
   32 EPSE(J)=EPST(I,J)-EPSP(I,J)
      DO33 J=I,4
      SIGMA(J)=0.0
      DO33K=1,4
   33 SIGMA(J)=SIGMA(J)+C(I,J,K)*EPSE(K)
C
      DC 8 J=1,4
      SIGMAP(J)=0.0
      DC 8 K=1,4
    8 SIGMAP(J)=SIGMAP(J)+C(I,J,K)*EPSP(I,K)
      DC 9 J=I,4
      NODE=NP(I,J)
      IF(NODE.EQ.0) GO TO 9
      NPR=NODE-NPOUT(ICLUST
      DC 10 K=1,4
   10 SIGPL(NPR,K)=SIGPL(NPR,K)+SIGMAP(K)
    9 CONTINUE
    7 CONTINUE
    6 CONTINUE
      WRITE(6,36)
   36 FORMAT(1H0,15HNODE POINT CATA//)
  102 IF(ISWTCH.EQ.1) GO TO 103
      READ(1) NPLOW,NPHIGH,N3,NUMCP,N5,N6,NUMNPB,(NADJNP(I),
     1NADJEL(I),(NPADJ(I,J),J=1,MXADJP),(STNPU(K,I),STNPW(K,I),
     2K=1,4),((STADU(K,I,J),STADW(K,I,J),K=1,4),J=1,MXADJP),I=1,NUMNPB)
C
      NLOW=NPLOW-NPOUT(ICLUS)
      NHGH=NPHIGH-NPOUT(ICLUS)
      DO 11 I=NLOW,NHGH
      DUM=NADJEL(I)
      DO 11 J=1,4
   11 SIGPL(I,J)=SIGPL(I,J)/DUM
C
  103 CONTINUE
      NLOW=NPLOW-NPOUT(ICLUS)
      NHGH=NPHIGH-NPOUT(ICLUS)
      DO 12 I=NLOW,NHGH
      NO=I+NPOUT(ICLUS)
      NUM=NADJNP(I)
      DO 13 J=1,4
      SIG(I,J)=STNPU(J,I)*U(NO)+STNPW(J,I)*W(NO)
      DC 14 K=1,NUM
      NODE=NPADJ(I,K)
   14 SIG(I,J)=SIG(I,J)+STADU(J,I,K)*U(NODE)+STADW(J,I,K)*W(NODE)
      SIG(I,J)=SIG(I,J)-SIGPL(I,J)
   13 CONTINUE
      DUM1=(SIG(I,1)-SIG(I,3))/2.
      DUM2=DUM1*DUM1+SIG(I,4)*SIG(I,4)
      IF(DUM2.GT. 0.0) GO TO 15
      RADIUS=0.0
      GO TO 16
   15 RADIUS=SQRT(DUM2)
   16 DUM3=(SIG(I,1)+SIG(I,3))/2.
      SIGMX(I)=DUM3+RADIUS
      SIGMN(I)=DUM3-RADIUS
      IF(DUM1.GE.0.0) GO TO 17
      THE=ATAN(SIG(I,4)/(-DUM1))
      THE=(PI-THE)/2.
```

```
      GC TO 20
   17 IF(DUM1.GT.0.0) GO TO 19
      IF(SIG(I,4).EQ.0.0) GO TO 18
      THE=PI/4.
      GO TO 20
   18 THE=0.0
      GC TO 20
   19 THE=0.5*ATAN(SIG(I,4)/DUM1)
   20 THETA(I)=THE*180./PI
   12 CONTINUE
      WRITE(6,21)
   21 FCRMAT(10H  NEW NODE,10X,6HU (IN),12X,12HSIGMAR (PSI),8X,
     112HSIGMAZ (PSI),8X,12HSIGMX  (PSI),9X,11HTHETA (DEG)/
     210H  OLD NCDE,10X,6HW (IN),12X,12HSIGMAT (PSI),8X,12HTAU    (PSI),
     38X,12HSIGMN  (PSI)//)
      DO 22 I=NLOW,NHGH
      NPNEW=I+NPCUT(ICLUS)
      NPOLD=NPTN(NPNEW)
C***********************************************************************
      IF(IMPBX.EC.1)WRITE(15)NPOLD,U(NPNEW),W(NPNEW)
C***********************************************************************
   22 WRITE(6,23) NPNEW,U(NPNEW),SIG(I,1),SIG(I,3),SIGMX(I),THETA(I),
     1            NPOLD,W(NPNEW),SIG(I,2),SIG(I,4),SIGMN(I)
   23 FCRMAT(I7,3X,1P5E20.5/I7,3X,1P4E20.5//)
      NMRCDS=NHGH-NLOW+1
  101 CCNTINUE
      REWIND 1
  100 CONTINUE
      REWIND IOJTAP
      REWIND INT
      REWIND 1
      RETURN
      END
```

# APPENDIX B - INTERPOLATION CODE FOR DETERMINING MPBX DISPLACEMENTS

## B.1 - Code Description

This code determines the displacement of desired points along multiple position borehole extensometers (MPBX) by interpolating between node point displacements determined in the static SLAM finite element code. The code is written entirely in FORTRAN IV and consists of a main program, and five subroutines.

The code accepts either punched card input, which is read via tape 5, or input stored on magnetic tape, read as tape 1, output of the static SLAM code. Output is printed via tape 6. The code is presently operational on the CDC 6500, using the Purdue MACE operating system.

## B.2 - Data Deck Setup

The following description of the data deck setup assumes that, in general, all numbers are right-oriented in their fields. Inclusion of the decimal point in floating point (real) numbers overrides the right-orientation requirement. Integer data are entered in 5-column fields while all floating point data are entered in 10-column fields.

| CARD | VARIABLE | FORMAT |
|---|---|---|
| 1.1 | ANAME | (18A4) |

ANAME = Problem descriptor to be printed as output, up to 72 characters.

| | | |
|---|---|---|
| 2.1 | ITAPE | (I5) |

ITAPE = Counter to indicate whether node point and element input is to be on punched cards or magnetic tape.

= 0, Node point and element input data are on punched cards.

= 1, Node point and element input data are on magnetic tape.

| | | |
|---|---|---|
| 3.1 | NUMNP, NUMEL | (2I5) |

NUMNP = Number of node points ($\leq$ 1600)

NUMEL = Number of elements

Note: Card 3.1 omitted if ITAPE = 1.

| | | |
|---|---|---|
| 4.1 | NPNUM, R, Z, U, V | (I5,2F10.3, 2E10.4) |

NPNUM = Node point number

R = Radial (horizontal) coordinate (ft) of node point

Z = Vertical coordinate (ft) of node point

U = Horizontal displacement (inches) of node point (positive to right)

V = Vertical displacement (inches) of node point (positive down).

Note: Card 4.1 Repeated NUMNP times if ITAPE = 0; card is omitted if ITAPE = 1

| | | |
|---|---|---|
| 5.1 | NUME, NPI, NPJ, NPK, NPL | (5I5) |

NUME = Element Number

NPI to NPL = Node numbers at vertices of rectangular element. NPI may be any node, but NPJ, NPK, NPL must be given in clockwise order around element starting from NPI. If NPL = 0, element is considered to be a triangle.

Note: Card 5.1 repeated NUMEL times if ITAPE = 0; card is omitted if ITAPE = 1.

| CARD | VARIABLE | FORMAT |
|---|---|---|
| 6.1 | NBX | (I5) |

NBX = Number of MPBX lines considered.

| | | |
|---|---|---|
| 6.2 | NMPBX, PBX (1), PBX (2), PBX (3), PBX (4) | (I5,4F10.2) |

NMPBX = MPBX identification number

PBX (1) = Radial (horizontal) coordinate (ft) of leftmost end of MPBX. If MPBX is vertical, this is the coordinate for the lower end.

PBX (2) = Vertical coordinate (ft) of leftmost end of MPBX.

PBX (3) = Radial (horizontal) coordinate (ft) of rightmost end of MPBX.

PBX (4) = Vertical coordinate (ft) of rightmost end of MPBX.

Note: Card 6.2 repeated NBX times.

## B.3 - Output

The output for each MPBX considered gives the MPBX identification number, coordinates of the end points, and displacements parallel to the MPBX line of points on the line. Displacements are determined for each point on the MPBX line where it intersects a line joining two adjacent node points.

Displacements along the MPBX line are defined to be positive if they occur in the direction from the leftmost end point toward the rightmost end point (irrespective of which end corresponds to the tunnel face). When the MPBX is vertical positive displacement corresponds to movement from the lower end point toward the higher end point.

```
      PROGRAM WHP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
      DIMENSION PBX(10,5),NP(1600,10),R(1600),Z(1600),
     1U(1600),V(1600),NADNP(1600)
      WRITE(6,5)
5     FORMAT(*1        DISPLACEMENTS ALONG MPBX LINES*//
     1*NOTE--DISPLACEMENTS ARE MEASURED POSITIVE IF IN */
     2*DIRECTION FROM LEFTMOST ENDPOINT TO RIGHTMOST ENDPOINT.*/
     3*IF MPBX IS VERTICAL,POSITIVE DISPLACEMENT */
     4*IS MEASURED IN DIRECTION FROM LOWEST TO HIGHEST POINT*)
 500  READ(5,6)ANAME
   6  FORMAT(18A4)
      WRITE(6,6) ANAME
      READ(5,7)ITAPE
      IF(ITAPE.EQ.1) GO TO 1000
      READ(5,7)NUMNP,NUMEL
 7    FORMAT(3I5)
C  READ IN NODE POINT DATA INCLUDING DISPLACEMENT
      READ(5,8)(NPNUM,R(NPN),Z(NPN),U(NPN),V(NPN),
     1NPN=1,NUMNP)
      GO TO 1010
 1000 REWIND 1
      READ(1)NUMNP,(R(I),Z(I),I=1,NUMNP),NUMEL
   8  FORMAT(I5,2F10.3,2E10.4)
C  READ IN ELEMENT DATA
C  READ IN SUBROUTINE
 1010 CALL GETNP(NUMNP,NUMEL,NADNP,NP,ITAPE)
      IF (ITAPE.EQ.0) GO TO 1020
      READ (1) (I,U(I),V(I),J=1,NUMNP)
 1020 DO 1015 I=1,NUMNP
      Z(I)=-Z(I)
 1015 V(I)=-V(I)
      READ(5,7) NBX
      DO 400 IBX=1,NBX
C  MPBX INPUT-- LEFTMOST ENDPOINT READ FIRST
      READ(5,9)NMPBX,(PBX(IBX,I),I=1,4)
 9     FORMAT(I5,4F10.2)
      WRITE(6,10)NMPBX,(PBX(IBX,I),I=1,4)
 10   FORMAT(//*  MPBX NO. = *I4*,   END POINTS - RLEFT = *
     1F8.2,*,  ZLEFT = *F8.2/33X*RRIGHT = *F8.2*,  ZRIGHT = *F8.2/)
      WRITE(6,12)
 12   FORMAT(5X,*R-COORD.*5X,*Z-COORD.*5X,*DISPLACMT.(IN.)*/)
      PBX(IBX,2)=-PBX(IBX,2)
      PBX(IBX,4)=-PBX(IBX,4)
C  CHECK IF MPBX IS VERTICAL
      IF(ABS(PBX(IBX,1)-PBX(IBX,3)).GT.0.01) GO TO 20
      PBX(IBX,5)=99999.0
      GO TO 25
C  CALCULATE SLOPE OF MPBX
 20   PBX(IBX,5)=(PBX(IBX,4)-PBX(IBX,2))/(PBX(IBX,3)-PBX(IBX
     1,1))
      IF(PBX(IBX,5).EQ.0.0)PBX(IBX,4)=PBX(IBX,4)+0.001
 25   CALL MAX(PBX(IBX,2),PBX(IBX,1),PBX(IBX,4),PBX(IBX,3),
     1PBXTOP,PBXBOT,RINTER,IXCODE)
      DO 200 NPN=1,NUMNP
      IF(NADNP(NPN).EQ.0) GO TO 200
      NADJP=NADNP(NPN)
      DO 300 IAP=1,NADJP
C  CHECK IF NODE LINE HAS BEEN USED
      IF(NP(NPN,IAP)-NPN) 300,300,30
 30   NPADJ=NP(NPN,IAP)
```

```
C  CHECK IF NODE LINE IS VERTICAL
      IF(ABS(R(NPN)-R(NPADJ)).GT.0.01) GO TO 40
      SLOPNP=99999.0
      GO TO 50
C  CALCULATE SLOPE OF NODE POINT LINE
 40   SLOPNP=(Z(NPADJ)-Z(NPN))/(R(NPADJ)-R(NPN))
C  CHECK FOR PBX PARALLEL TO NODE PT. LINE
 50   IF(ABS(SLOPNP-PBX(IBX,5)).LT.0.01) GO TO 300
      IF(SLOPNP.NE.99999.0) GO TO 85
      RINTER=R(NPN)
      IF(RINTER.LT.PBX(IBX,1).OR.RINTER.GT.PBX(IBX,3))
     1GO TO 300
      CALL MAX(Z(NPN),R(NPN),Z(NPADJ),R(NPADJ),ZTOP,ZBOT,RBO
     1T,NPCODE)
      ZINTER=PBX(IBX,2)+PBX(IBX,5)*(RINTER-PBX(IBX,1))
      IF(ZINTER.LT.ZBOT.OR.ZINTER.GT.ZTOP) GO TO 300
      GO TO (60,70),NPCODE
 60   NTOP=NPN
      NBOT=NPADJ
      GO TO 80
 70   NTOP=NPADJ
      NBOT=NPN
 80   CALL INTERP(ZBOT,ZTOP,ZINTER,PBX(IBX,5),U(NBOT),V(NBOT
     1),U(NTOP),V(NTOP),DISPL)
      GO TO 350
 85   CALL MAX(R(NPN),Z(NPN),R(NPADJ),Z(NPADJ),RRIGHT,RLEFT,
     1ZLEFT,NPCODE)
      IF(PBX(IBX,5).EQ.99999.0) GO TO 90
      RINTER=(PBX(IBX,2)-ZLEFT+SLOPNP*RLEFT-PBX(IBX,5)*PBX(I
     1BX,1))/(SLOPNP-PBX(IBX,5))
      IF(RINTER.LT.PBX(IBX,1).OR.RINTER.GT.PBX(IBX,3)) GOTO 300
 90   IF(RINTER.LT.RLEFT.OR.RINTER.GT.RRIGHT)GO TO 300
      ZINTER=ZLEFT+SLOPNP*(RINTER-RLEFT)
      IF(ZINTER.LT.PBXBOT.OR.ZINTER.GT.PBXTOP) GO TO 300
      GO TO (100,110),NPCODE
 100  NLEFT=NPADJ
      NRIGHT=NPN
      GO TO 120
 110  NLEFT=NPN
      NRIGHT=NPADJ
 120  CALL INTERP(RLEFT,RRIGHT,RINTER,PBX(IBX,5),U(NLEFT),V(
     1NLEFT),U(NRIGHT),V(NRIGHT),DISPL)
 350  ZINTER=-ZINTER
      WRITE(6,13) RINTER,ZINTER,DISPL
 13   FORMAT(3X,F8.2,5X,F8.2,5X,E12.4)
 300  CONTINUE
 200  CONTINUE
 400  CONTINUE
      STOP
      END
C
C
C
      SUBROUTINE GETNP(NUMNP,NUMEL,NADJNP,NPADJ,ITAPE)
      DIMENSION NPADJ(1600,10),NADJNP(1600)
      MXADJP=8
C
      DO 5 I=1,NUMNP
      NADJNP(I)=0
      DO 5 J=1,MXADJP
```

```
    5 NPADJ(I,J)=0
C
      DO 7 M=1,NUMEL
      IF(ITAPE.EQ.1) GO TO 6
      READ(5,4)NUME,NPI,NPJ,NPK,NPL
      GO TO 7
    6 READ(1) NUME,NPI,NPJ,NPK,NPL
    7 CALL ADJNP(NUMNP,NPADJ,NUME,NPI,NPJ,NPK,NPL)
C
      CALL VADJNP(NADJNP,NUMNP,NPADJ)
  4   FORMAT(5I5)
      RETURN
C
      END
C
C
C
      SUBROUTINE ADJNP(NUMNP,NPADJ,NUME,
     1NPI,NPJ,NPK,NPL)
      DIMENSION NPADJ(1600,10),NA(4)
C
C**** FORM TABLE OF ADJACENT NODAL POINTS
C     MXADJP=MAX. NO. OF ADJACENT NODAL POINTS ALLOWED
C     NUMNP =NO. OF NODE POINTS
C     NPADJ =ADJACENT NODE POINT NUMBER
C     NPI   =ELEMENT NODE POINT I
C     NPJ   =ELEMENT NODE POINT J
C     NPK   =ELEMENT NODE POINT K
C     NPL   =ELEMENT NODE POINT L, IF = 0 , TRIAN ELEMENT
C     NUME  =ELEMENT NUMBER BEING CONSIDERED
C     NOTE- TABLE ASSUMED TO BE ALREADY ZEROED OUT
C
      MXADJP=8
      NA(1)=NPI
      NA(2)=NPJ
      NA(3)=NPK
      NA(4)=NPL
      ICOUNT=1
    9 NPNUM=NA(1)
      MX=NA(2)
      JCOUNT=1
    5 DO 1 I=1,MXADJP
      J=I
      IF(NPADJ(NPNUM,I).EQ.MX) GO TO 2
      IF(NPADJ(NPNUM,I).EQ.0)  GO TO 3
    1 CONTINUE
      WRITE (6,101)  NUME,NPNUM,MX,(NPADJ(NPNUM,I),I=1,MXADJP)
      CALL EXIT
C
    3 NPADJ(NPNUM,J)=MX
    2 JCOUNT=JCOUNT+1
      IF(JCOUNT.GT.3) GO TO 4
      IF(JCOUNT.GT.2) GO TO 102
      MX=NA(3)
      IF(NPL.EQ. 0 ) GO TO 5
  102 MX=NA(4)
      GO TO 5
C
    4 GO TO (6,7,8,103),ICOUNT
    6 ICOUNT=2
```

```
      NA(1)=NPJ
      NA(2)=NPK
      NA(3)=NPL
      NA(4)=NPI
      GO TO 9
C
    7 ICOUNT=3
      NA(2)=NPL
      NA(3)=NPI
      NA(1)=NPK
      NA(4)=NPJ
      GO TO 9
C
    8 ICOUNT=4
      NA(1)=NPL
      IF(NPL.EQ. 0 ) GO TO 103
      NA(2)=NPI
      NA(3)=NPJ
      NA(4)=NPK
      GO TO 9
C
  101 FORMAT(43HERROR IN FORMING ADJACENT NODAL POINT ARRAY/
     121H ELEMENT NUMBER      =,I5/19HNODE POINT NUMBER =,I5/
     217H ADJA NODE POINT=,I5//15H NPADJ(NPNUM,I)/(21X,I5))
C
  103 RETURN
C
      END
C
C
C
      SUBROUTINE VADJNP(NADJNP,NUMNP,NPADJ)
      DIMENSION NADJNP(1600),NPADJ(1600,10)
C
C**** FORM VECTOR INDICATING THE NUMBER OF ADJACENT NODE POINTS
C     AT EACH NODE POINT
C
C     MXADJP=MAX. NO. OF ADJACENT NODE POINTS ALLOWED
C     NADJNP=NO. OF ADJACENT NODE POINTS AT EACH NODE POINT
C     NUMNP =NO. OF NODE POINTS
C     NPADJ =ADJACENT NODE POINT NUMBER
      MXADJP=8
      DO 12 M=1,NUMNP
      DO 10 I=1,MXADJP
      J=I
      IF (NPADJ(M,I).EQ.0) GO TO 11
   10 CONTINUE
      NADJNP(M)=MXADJP
      GO TO 12
   11 NADJNP(M)=J-1
   12 CONTINUE
      RETURN
C
      END
C
C
C
      SUBROUTINE MAX(A1,B1,A2,B2,AMAX,AMIN,BMIN,N)
      IF(A1-A2)10,20,20
 10   AMAX=A2
```

```
      AMIN=A1
      BMIN=B1
      N=2
      RETURN
 20   AMAX=A1
      AMIN=A2
      BMIN=B2
      N=1
      RETURN
      END
C
C
C
      SUBROUTINE INTERP(AMIN,AMAX,AINTFR,SLOPE,UMIN,VMIN,UMA
     1X,VMAX,DISPL)
      IF(ABS(AMIN-AMAX).GT.0.01) GO TO 20
      FACTOR=0.0
      GO TO 30
 20   FACTOR=(AINTFR-AMIN)/(AMAX-AMIN)
 30   UINTER=UMIN+FACTOR*(UMAX-UMIN)
      VINTER=VMIN + FACTOR*(VMAX-VMIN)
      ANGLE=ATAN(SLOPE)
      DISPL=UINTER*COS(ANGLE)+VINTER*SIN(ANGLE)
      RETURN
      END
```